NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

AN EXAMINATION OF THE ABILITY OF OCEAN OBERVATORY SYSTEMS TO DETERMINE MERCHANT SHIP DIRECTION AND DRAFT

by

Richard S. Hunt

September 2013

Thesis Advisor:	Daphne Kapolka
Second Reader:	Kevin B. Smith

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704–0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202–4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704–0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2013	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE AN EXAMINATION OF THE ABILITY OF OCEAN OBERVATORY SYSTEMS TO DETERMINE MERCHANT SHIP DIRECTION AND DRAFT	**5. FUNDING NUMBERS**
6. AUTHOR(S) Richard S. Hunt	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943–5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number ____N/A____.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

The objective of this thesis was to analyze data obtained from a network of Ocean Bottom Seismometers to determine if it could be used to provide detailed information regarding merchant vessels such as their direction and draft. The sensors were located in the Strait of Juan de Fuca and collected data from August–September, 2009. The hydrophone and three orthogonal seismometer channels were beamformed in MATLAB as a vector sensor in an attempt to get bearing data on a passing ship. Frequencies were limited to about 80Hz due to the low sampling frequency. A Lloyd's mirror pattern from the ship's broadband noise was visible in the lofargrams from all four channels during this transit. The Lloyd's mirror pattern was compared qualitatively with theoretical predictions from ray theory as well as with transmission loss predictions from the parabolic equation model run in PC-IMAT. Vector sensor beamforming proved unsuccessful due to the lack of coherence and erratic phase differences among the sensors. This erratic behavior is probably due to multipath effects. Both ray theory and PC-IMAT models show promise for exploiting the Lloyd's mirror patterns. The expected interference patterns show a clear dependence on range and draft.

14. SUBJECT TERMS acoustics, interference, vector sensor, Lloyd's mirror, lofargram	**15. NUMBER OF PAGES** 89
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

THIS PAGE INTENTIONALLY LEFT BLANK

AN EXAMINATION OF THE ABILITY OF OCEAN OBERVATORY SYSTEMS TO DETERMINE MERCHANT SHIP DIRECTION AND DRAFT

Richard S. Hunt
Lieutenant, United States Navy
B.S., University of Florida, 2005

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING ACOUSTICS

from the

NAVAL POSTGRADUATE SCHOOL
September 2013

Author: Richard S. Hunt

Approved by: Daphne Kapolka
Thesis Advisor

Kevin B. Smith
Second Reader

Daphne Kapolka
Chair, Engineering Acoustics Academic Committee

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

The objective of this thesis was to analyze data obtained from a network of Ocean Bottom Seismometers to determine if it could be used to provide detailed information regarding merchant vessels such as their direction and draft. The sensors were located in the Strait of Juan de Fuca and collected data from August–September, 2009. The hydrophone and three orthogonal seismometer channels were beamformed in MATLAB as a vector sensor in an attempt to get bearing data on a passing ship. Frequencies were limited to about 80Hz due to the low sampling frequency. A Lloyd's mirror pattern from the ship's broadband noise was visible in the lofargrams from all four channels during this transit. The Lloyd's mirror pattern was compared qualitatively with theoretical predictions from ray theory as well as with transmission loss predictions from the parabolic equation model run in PC-IMAT. Vector sensor beamforming proved unsuccessful due to the lack of coherence and erratic phase differences among the sensors. This erratic behavior is probably due to multipath effects. Both ray theory and PC-IMAT models show promise for exploiting the Lloyd's mirror patterns. The expected interference patterns show a clear dependence on range and draft.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AIS	Automatic Identification System
CPA	Closest Point of Approach
CPA	Closest Point of Approach
DMC	IRIS Data Management Center
GPS	Global Positioning System
IRIS	Incorporated Research Institute for Seismology
mseed	MiniSEED
NSF	National Science Foundation
OBS	Ocean Bottom Seismometer
OBSIP	Ocean Bottom Seismography Institute Pool
OOS	Ocean Observing System
OOSSG	Ocean-Observing System Security Group
PC-IMAT	Personal Computer–Interactive Multisensor Analysis Training
SAC	Seismic Analysis Code
SNR	Signal to Noise Ratio
SSP	Sound Speed Profile
WHOI	Woods Hole Oceanographic Institute

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would first like to thank my wife, Nikki, and my family for all of their help and understanding during my time working on this thesis. Without the help and support this would not have been possible. I would also like to thank my thesis advisor, Daphne Kapolka, for all of her help getting this thesis together and reviewed. You were always willing to give me time and help during the entire process. I couldn’t have done this without the help of everyone involved. Thank you all.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. CONCEPT

The purpose of this thesis was to analyze acoustic data generated by a set of sensors chosen from the infrastructure of Ocean Observing Systems (OOS) to determine if it is possible to deduce detailed information, such as direction of travel and draft, on passing merchant ships. This issue is of particular use to the United States Navy, especially in high merchant traffic areas. This thesis continues work from a thesis completed in 2012 [1], which developed an automated detector to identify ship tonals in OOS data. Around the world there are a growing number of OOS, which are networks of underwater sensors of varying capabilities in use to study the ocean environment. These sensors have different capabilities, and the use of them can help to understand and predict events originating in the ocean including geologic and weather events. The data that the sensors provide, however, can also be used for other purposes. One of the types of sensors in use is the Ocean Bottom Seismometer (OBS). OBS have multiple sensors onboard to record data, and this thesis focuses on the data generated by the hydrophones and seismometers on these devices. The first goal of this thesis was to beamform the data retrieved, using the particle velocity and pressure, to examine whether the direction of the merchant vessel(s) could be determined. The second goal of this thesis was to determine whether Lloyd's mirror patterns could be exploited to establish the draft of the merchant vessel(s).

B. ORGANIZATION

This thesis will start out in the next chapter with more details on the history of ocean measurements and of Ocean Bottom Seismometers in particular. This background will also discuss previous work on beamforming seismic sensors and the use of Lloyd's mirror patterns in ocean acoustics. The details of the theory used in this thesis will be discussed in the next chapter. A brief discussion of the expected particle velocity from a distant acoustic contact will be presented. This is followed by the equations used to beamform particle velocity and pressure to determine contact bearing. To understand the interference patterns which are so frequently seen in sonar data, the two major methods used to model acoustic propagation in the ocean, ray theory and mode theory, are briefly discussed. The theory chapter concludes with a more detailed analysis of both short range (ray theory) and long range (mode theory) interference patterns. The next chapter consists of the methodology used to obtain and analyze the data as well as a discussion of the results. The final chapter offers conclusions and recommendations for future work. The MATLAB code used in this thesis and the calibration data for the OBS sensors used are contained in the appendices.

C. SUMMARY OF RESULTS

Attempts to determine the bearing of a loud merchant ship using the inherent directionality of the seismometer channels were unsuccessful. The measured bearings changed in a seemingly random fashion despite high signal to noise (SNR). This result is in agreement with other studies. The combination of multiple acoustic paths, unknown coupling between surface and waterborne acoustic waves, and unrelated seismic signals combine to distort the amplitude and phase of the particle velocity relative to the hydrophone and make it impossible to determine the directionality of the acoustic signal.

The use of the striations in the Lloyd’s mirror pattern to determine vessel draft shows much more promise. Models of transmission loss run in PC-IMAT (Personal Computer–Interactive Multisensor Analysis Training) show a distinct difference in the interference pattern as a function of depth and range. Further work is needed to determine the extent to which these patterns can be exploited.

II. BACKGROUND

A. OCEAN BOTTOM SEISMOMETERS

Ocean Observatory Systems (OOS) systems began to be implemented decades ago. The first global system began even before current technology included electronic systems to generate data. In the 1800s, Matthew Fontaine Maury planned to organize information from captain’s logbooks and copies of their maps. This helped to set the common standards and formats for data collection and initiate a free exchange of data and information for the global good. By 1990, this had expanded and adapted using current technology to a global system of coordinated observations and data communications. With the oceans covering the majority of the globe, and very little of that area being used, it provides the largest source for the continued growth and expansion of global monitoring networks. With growing concerns over worldwide climate changes, energy usage and independence, and natural disasters, observation of ocean events and the ocean environment has become of increasing concern. In January 2010, only an estimated 62% of the future planned networks had been completed [2]. As the locations and numbers of these systems grow, the abilities of the technology improve as well. Due to this growth and the future expansion, the sensors are expected to gain popularity and usage, and the systems will be more valuable as the capabilities increase.

One of the types of sensors used in OOSs is the Ocean Bottom Seismometer. These sensors are primarily used to measure movement in the Earth’s crust. With about 90% of earthquakes occurring underwater, the OBS was developed to gather data from these earthquakes to help researchers working to detect, understand, and predict earthquakes [3]. An OBS consists of a sphere which contains the sensors, batteries, and other equipment, with an anchor to keep it securely on the ocean floor [4]. Generally, these devices are designed to be deployed and recovered from almost any research vessel. Scientists submit proposals to do research using the equipment available from the U.S. National Ocean Bottom Seismography Institute Pool (OBSIP) established by the National Science Foundation (NSF).

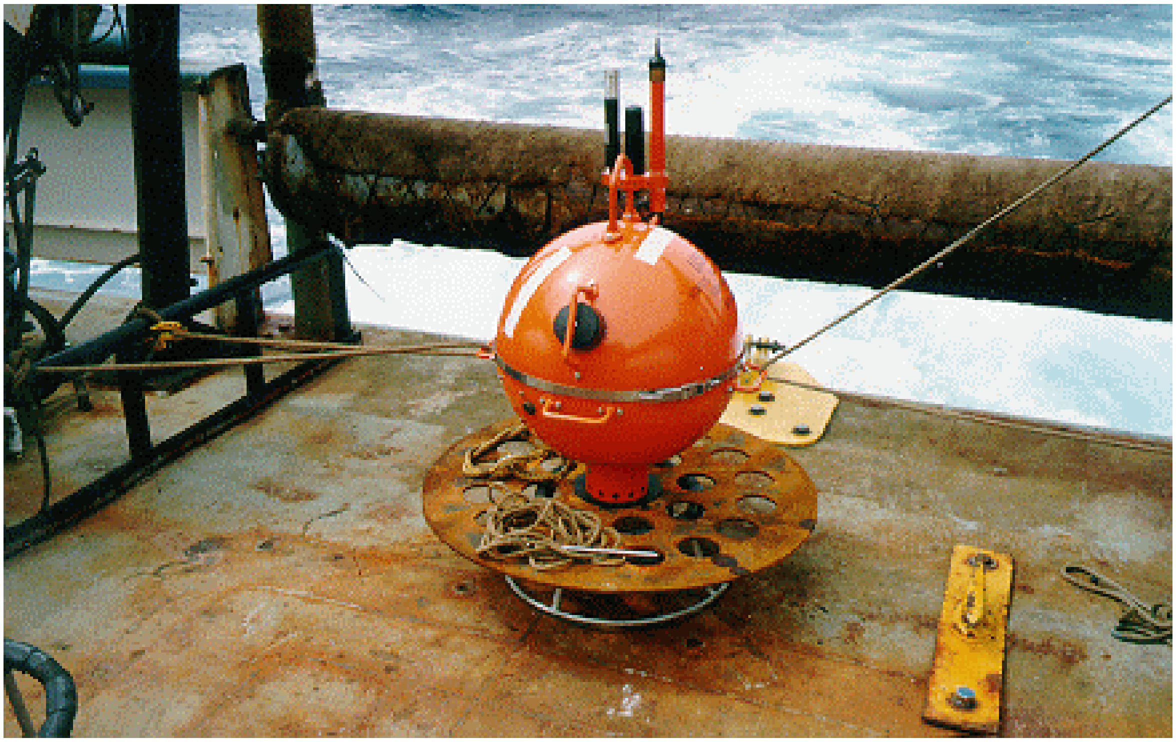


Figure 1. Picture of an OBS. From [5].

The data for this thesis was obtained from a set of 23 sensors approximately 200 nm west of the Strait of Juan de Fuca located in Washington. These sensors obtained their data during a period from August 22, 2009 to September 16, 2009. The data is archived by the Incorporated Research Institute for Seismology (IRIS), with the title "Strait of Juan de Fuca Passive Source Tomography Experiment" [6]. They are designated as the "YN" network on the IRIS website.

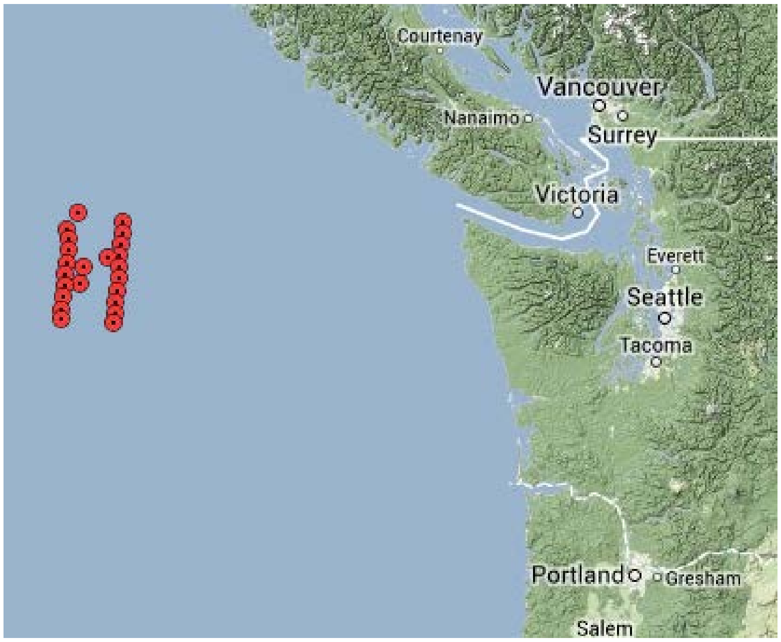


Figure 2. Location of the Strait of Juan de Fuca Passive Source Tomography Experiment Sensors. From [6].

The OBS instruments used for the Juan de Fuca experiment were provided by the Woods Hole Oceanographic Institute (WHOI) equipment pool as part of OBSIP. They use the Quanterra Q330 multichannel broadband, high resolution seismic system for data acquisition, telemetry, and sensor control. The Q330 has an adjustable sample rate, but the channels used for this thesis were sampled at 200 Hz. It is not entirely clear from the IRIS website, but it seems likely that the geophones and the hydrophone data were collected by two different Q330s. If they were collected by different Q330s, one may have provided the master clock signal. In any case, the Q330 time base is locked to GPS before deployment [6], [7]. The time drift reported by one source is estimated to be less than 32ns per s [8].

The seismometers or geophones in the WHOI OBS10 are Geospace GS-11Ds. They consist of a mass suspended by a spring between the poles of a magnet which is used to measure the voltage produced by movement of the OBS. The gain of the GS-11D is flat from 10–80Hz with a value of 1.02×10^9 *counts/m/s*. The phase response is fairly flat between 10–40Hz with a value of about 38^0. There are three orthogonal geophones in the OBS designated EL1, EL2, and ELZ. They face east, north, and vertically downward respectively. There is no apparent difference in the calibration curves for the three orthogonal geophone channels, and the value provided for the sensitivity gain is identical to six significant figures. These channels are sensitive to any motion including surface waves generated by earthquakes and motion induced by acoustic waves traveling through the water. The calibration curve of EL1 is included in Appendix A.

The hydrophone used for the experiment was the HTI-90-U. It is a high performance, high sensitivity hydrophone with low self-noise. This hydrophone has a maximum operating depth of 6,096 meters and a frequency response ranging from 2Hz to 20 kHz [9]. The gain of the hydrophone was also flat with a value of $349\,counts/Pa$from 10–80Hz and a fairly flat phase of 30^0 from 10–40Hz. The calibration curve for the hydrophone is also provided in Appendix A.

B. BEAMFORMING SEISMIC SENSORS

A hydrophone measures the pressure from a sound wave. Unless the hydrophone is directional, this will lead to ambiguity in the direction the signal came from which would require an array of hydrophones to resolve. The lower the frequency of the signal, the larger the array needs to be to determine the direction. In contrast, a vector sensor measures the particle velocity as well as the pressure. This can give the direction a signal is coming from as well as the same information a hydrophone is able to receive. The relatively compact size of a vector sensor makes it well suited for use in unmanned underwater vehicles and sensor arrays [10]. Much work has been done in recent years to examine the advantages and limitations of vector sensors.

Since the OBS equipment has three orthogonal seismometers in addition to a hydrophone, it has the essential components of a vector sensor. In fact, the seismometer data is given in units of particle velocity. These seismometers are primarily designed to sense the motion of waves traveling along the ocean bottom, i.e., seismic signals. However, they are also sensitive to motion due to acoustic signals. To the extent that the velocity measured by the seismometer channels is equal to the particle velocity from an acoustic signal, the data obtained from the seismometers and hydrophone can be processed to determine the directionality of the signal.

A simple beamforming algorithm for vector sensors was provided by [11]. This algorithm was successfully applied to beamform a linear array consisting of a conventional microphone and two Microflown in-air vector sensors after performing an in-situ calibration of the particle velocity channels in the frequency domain [12]. The array in this study was located in an anechoic chamber. A follow-on study [13] showed problems in a more realistic in-air environment. Signals received from aircraft flying overhead showed evidence of interference patterns from reflections of the sound off the concrete. These reflections prevented the beamformer from accurately determining the aircraft bearing.

The underwater realm is even more complicated. A number of papers have been published on seismometer signals measured on the ocean bottom. One early paper discusses how shear waves trapped in the bottom sediment layer can cause ringing in the seismic signals which are not observed in the hydrophone data [14]. This study captures the biggest problem with trying to use ocean bottom seismometers and their associated hydrophones as acoustic vector sensors. The acoustic particle velocity is not the only signal these sensors will detect. In fact, for researchers interested in seismic signals due to earthquakes and other movements in the Earth's crust, the signal from acoustic signals constitute unwanted noise. For example, [15], [16] discuss the sensitivity of ocean bottom seismometers to waterborne signals and point out the necessity of knowing the transfer function between the acoustic signal and the seismometer output. They state that the vertical component of the seismometer is expected to be coupled more strongly to the waterborne sound. Duennebier and Sutton [15] point out that better coupling might be achieved for the horizontal seismometer signals if the OBS is buried and has a density equal to the sediment density. This is consistent with the general design goal for vector sensors of neutral buoyancy. Osler and Chapman [16] point out that due to the complexity of the types of waves which can be detected on a seismic sensor, the transfer function between waterborne and surface waves depends on angle of incidence in addition to other factors. Smith [17] elaborates on the idea that the transfer functions have an angular dependence. This article specifically discusses the impediment to vector sensor beamforming due to the fact that the induced shear waves are so strongly dependent on a wide variety of factors.

C. LLOYD’S MIRROR PATTERNS

In the early 1800s, Humphrey Lloyd conducted an experiment with light from a monochromatic slit source. The light reflected off a glass surface and onto a screen. The direct light from the source and the reflected light interfered with each other causing what became known as a Lloyd’s mirror effect [18]. A similar effect was also discovered in ocean acoustical signals.

> When the sea surface is not too rough, it creates an interference pattern in the underwater sound field. This pattern is caused by constructive and destructive interference between the direct and surface reflected sound and is called the Lloyd’s mirror, or image interference, effect. [19]

In 1983, Hudson showed how the Lloyd’s mirror pattern could be used to calculate the velocity, depth and range of a submerged contact at its closest point of approach (CPA). This work used a straight-line approximation of ray theory to calculate these tactically valuable quantities [20].

Ray theory seeks to understand the acoustic field as the sum of the sound reaching the receiver along different paths. Sound rays refract according to the changing sound speed, but the speed of sound changes very slowly in the ocean. Therefore, these paths are fairly linear over short ranges. If the sound which travels between a source and a receiver is limited to a few paths, it is computationally fairly simple to apply ray theory. If the sound traveling along the different paths is coherent—meaning that there is a constant phase difference between them—interference may be noticeable in the received signal. Details of the Lloyd’s mirror pattern analysis using ray theory will be provided in the next chapter where theory is discussed.

D. THE WAVEGUIDE INVARIANT

An alternative means of understanding the propagation of sound in the ocean is to use normal mode theory. This approach seeks to decompose the acoustic field into a sum of orthogonal solutions to the wave equation. Interference patterns in acoustic data can be understood in terms of mode theory as well. Drawing on earlier work [21], Brekhovskikh and Lysanov [22] point out how the interference of closely spaced modes will yield large interference periods while widely spaced modes yield shorter interference periods. A parameter called the “waveguide invariant” sums up the dispersion characteristics of a waveguide. Brekhovskikh and Lysanov [22] show that the waveguide invariant has a value $\beta \cong 1$ when the contributing modes are at have very low grazing angles. In contrast, the waveguide invariant for a surface sound channel is derived as $\beta \cong -3$. More details on the significance of these numbers are included in the next chapter.

D’Spain and Kuperman [23] distinguish between a short range interference pattern between a direct and reflected path—which they refer to as the Lloyd’s mirror effect—and the long range interference due to mode interference. They present an example of a 200m deep environment in which sound speed increases from a surface speed of 1450m/s to a bottom speed of 1500m/s. At short range the Lloyd’s mirror pattern is insensitive to water conditions since the paths are limited and fairly straight. At long ranges in this environment, the interference pattern changes its appearance, and they show how these changes can be predicted even in a range dependent environment using the waveguide invariant. Their primary focus was on using the observed interference patterns or striations to understanding the properties of the waveguide.

Thode [24] published an article the following year turning the use of the waveguide invariant to a determination of source range. The proposed technique requires a broadband source at a known range to a vertical array. Narasimhan and Krolik [25] had previously pointed out that such techniques for passive ranging would be sensitive to uncertainty in the environmental parameters, but the error could be greatly reduced by using the statistics of the environmental uncertainty. Zurk and Tracey [26] suggested an improvement to Thode’s technique by “depth shifting” the broadband source used to obtain additional depth dependent transfer functions between the source and receiver which could be used to refine the estimates made from acoustic measurements.

Several other papers [27], [28], [29], [30] examine the improvements which can be made in determining source range, waveguide properties, and perhaps even (roughly) depth by using horizontal arrays as opposed to a single receiver. In fact, Rouseff and Leigh [28] propose a technique for striation-based beamforming to estimate the waveguide invariant itself.

Brooks et al. [31] discuss techniques to determine the numerical value of the waveguide invariant from the observed striations in spectrograms. The basic idea is to find the line on the spectrogram which minimizes the variance in the sound intensity. Specifically, the use of spectrograms from passing ships of opportunity is promoted as a convenient source.

Interestingly, although D'Spain and Kuperman distinguish between the near-field Lloyd's mirror pattern and the long range interference governed by the waveguide invariant, [32] shows that the behavior of the striations in the spectrogram or lofargam is identical in cases where the waveguide invariant is equal to one.

III. THEORY

The details of the theory used to analyze the data obtained from the OBS sensors will be provided in this chapter. First, the basic theory behind beamforming vector sensors to determine source directionality will be presented. This includes the expected relationship between the pressure and particle velocity channels as well as the beamforming algorithm used by [11]. Attention is then turned to the theory required to answer the question of whether the interference patterns observed in the lofargrams can be used to estimate ship draft. Starting with simple straight line ray paths, both long and short range solutions for the expected interference are provided. This is then contrasted with the mode theory approach. The waveguide invariant parameter is presented and its predictions compared with those obtained from ray theory.

A. VECTOR SENSOR BEAMFORMING

To understand how the directionality of a sound source can be determined from a combination of hydrophone and three orthogonal velocity sensors, it is typical to do the analysis in the frequency domain and then use Fourier analysis to extend the result to more realistic signals consisting of multiple frequencies. In this work, complex numbers are denoted by a circumflex, and vectors are in bold. Using the same line of reasoning as [33], the far-field pressure of a single frequency acoustic wave can be written as

$$\hat{p}\left(r,t\right)=p_0 e^{i\left(\omega t-\mathbf{k}\cdot\mathbf{r}\right)} \tag{0.0}$$

where p_0 is the amplitude, ω is the angular frequency, **k** is the propagation vector of the acoustic wave, and **r** is the position where the wave is measured. This is the equation for a plane wave, and the acoustic particle velocity of the wave in the x-, y-, and z-directions depend on the propagation vector and the specific acoustic impedance of the medium. The impedance depends on the density, ρ, and the speed of sound, c. In terms of these quantities, the three components of particle velocity are given by

$$\hat{u}_x = \frac{\hat{p}}{\rho c}\left(\frac{\mathbf{k}}{k}\cdot\mathbf{n}_x\right) = \frac{\hat{p}}{\rho c}\frac{k_x}{k}, \quad \hat{u}_y = \frac{\hat{p}}{\rho c}\left(\frac{\mathbf{k}}{k}\cdot\mathbf{n}_y\right) = \frac{\hat{p}}{\rho c}\frac{k_y}{k},$$

and $$\hat{u}_z = \frac{\hat{p}}{\rho c}\left(\frac{\mathbf{k}}{k}\cdot\mathbf{n}_z\right) = \frac{\hat{p}}{\rho c}\frac{k_z}{k}, \qquad (0.0)$$

where $\mathbf{n}_\mathbf{x}$, $\mathbf{n}_y$, and $\mathbf{n}_z$ are the unit vectors along the axes.

Scaling each component of particle velocity by the impedance yields components which would have the same amplitude as the pressure if not reduced by the dot product, between the unit vector in the direction of the wave and the axis direction. In this sense, they are now commensurate with the pressure signal. For example, if the wave is traveling in the x-direction, $\hat{u}_y = \hat{u}_z = 0$ and $\rho c u_x = p$. So the experimental value of the particle velocity is multiplied by the impedance prior to beamforming.

It is the relative amplitude of the particle velocity components and their phase relative to pressure which determine the direction of the sound. To make the beamformer most sensitive to sounds coming from a source in the direction $\mathbf{n}_\mathbf{s}$, the beamformer needs to weight the various particle velocity channels appropriately. This steer direction, $\mathbf{n}_\mathbf{s}$, can be written in terms of an azimuthal angle, ϕ_s, and polar angle, θ_s, as

$$\mathbf{n}_s = \sin\theta_s \cos\phi_s \mathbf{n}_x + \sin\theta_s \sin\phi_s \mathbf{n}_y + \cos\theta_s \mathbf{n}_z . \qquad (0.0)$$

Define a weighting vector in terms of the steer direction as

$$\mathbf{w} = -\left[\left(\mathbf{n}_s \cdot \mathbf{n}_x\right)\mathbf{n}_x + \left(\mathbf{n}_s \cdot \mathbf{n}_y\right)\mathbf{n}_y + \left(\mathbf{n}_s \cdot \mathbf{n}_z\right)\mathbf{n}_z\right]. \qquad (0.0)$$

The negative sign is required because sound coming from a positive direction travels in a negative direction. The particle velocity beamformer output is then given in terms of this weighting vector as

$$B^{(pu)}\left(\phi_s, \theta_s\right) = \left|\left(\rho c \mathbf{w} \cdot \hat{\mathbf{u}} + \hat{p}\right)\right|^2 . \qquad (0.0)$$

To show how this works in general, consider sound propagating with propagation vector,

$$\mathbf{k} = k_x\mathbf{n_x} + k_y\mathbf{n}_y + k_z\mathbf{n}_z. \qquad (0.0)$$

When the beamformer is steered towards the source, the steer direction can be written in terms of the propagation vector as

$$\mathbf{n_s} = -\frac{\mathbf{k}}{k} = -\left(\frac{k_x}{k}\mathbf{n_x} + \frac{k_y}{k}\mathbf{n}_y + \frac{k_z}{k}\mathbf{n}_z\right). \qquad (0.0)$$

This makes the weighting vector, $\mathbf{w}$

$$\mathbf{w} = \left[\frac{k_x}{k}\mathbf{n}_x + \frac{k_y}{k}\mathbf{n}_y + \frac{k_z}{k}\mathbf{n}_z\right] \qquad (0.0)$$

Substituting the general expression for particle velocity given by (3.2) into the beamformer gives

$$\begin{aligned} B^{(pu)}(\phi_s, \theta_s) &= \left|\rho c\left[\frac{k_x}{k}\mathbf{n}_x + \frac{k_y}{k}\mathbf{n}_y + \frac{k_z}{k}\mathbf{n}_z\right]\Box\left(\frac{\hat{p}}{\rho c}\frac{k_x}{k}\mathbf{n_x} + \frac{\hat{p}}{\rho c}\frac{k_x}{k}\mathbf{n_x} + \frac{\hat{p}}{\rho c}\frac{k_x}{k}\mathbf{n_x}\right) + \hat{p}\right|^2 \\ &= \left|\left(\frac{k_x^2 + k_y^2 + k_z^2}{k^2}\right)\hat{p} + \hat{p}\right|^2. \end{aligned} \qquad (0.0)$$

Since $k_x^2 + k_y^2 + k_z^2 = k^2$, the maximum output of the beamformer is equal to

$$B^{(pu)}(\phi_s, \theta_s) = |2\hat{p}|^2 = 4p_0^2. \qquad (0.0)$$

The maximum output occurs when the beamformer is steered in the source direction so that $\mathbf{w} \cdot \hat{\mathbf{u}} = \hat{u}$. If the beamformer is steered in a direction which is orthogonal to $\hat{\mathbf{u}}$, the output of the beamformer is p_0^2. The theoretical output is zero when steered in the opposite direction to the source.

The analysis presented so far assumes that there are no reflections or other multi-paths whereby acoustic energy can travel from the source to the receiver and arrive with a different phase from the direct path. It is clear from the first beamformer equation that if the particle velocity channels are not either in phase or 180^0 out of phase with the pressure signal, the output of the beamformer will be degraded.

B. LONG RANGE RAY THEORY ANALYSIS OF LLOYD'S MIRROR PATTERNS

Ray theory is generally easier to apply at short ranges where there are a limited number of paths or eigenrays connecting the source and receiver. Ray paths are predicted on the basis of the refraction of sound due to the changing sound speed. The contributions from multiple paths are added either incoherently or coherently depending on whether the phase difference between the paths is random or not.

If the sea surface is not too rough, the direct and reflected path of the sound from a source to a receiver will generally be coherent enough to create an interference pattern known as the Lloyd's mirror pattern. When the depth of the source and receiver are small compared to the range, these two paths will have an approximate path length difference

$$\Delta r = \frac{2hd}{r} \tag{0.0}$$

where *h* is the receiver depth, *d* is the source depth, and *r* is the horizontal distance between the source and the receiver. It is important to note that the range still has to be short enough to justify the assumption that the ray paths are straight and to ensure that only the direct and surface reflected paths make significant contributions to the received signal. The validity of these assumptions depends primarily on the depths of the source, receiver, and water column.

Due to the 180^0 phase shift of surface reflected waves, nulls will occur at integral number of wavelengths between the direct and reflected sound paths. The range to nulls can be expressed as

$$r_n = \frac{2hd}{n\lambda} = \frac{2hd}{nc} f_n . \tag{0.0}$$

As a contact passes through CPA the geometry can be shown by Figure 3.

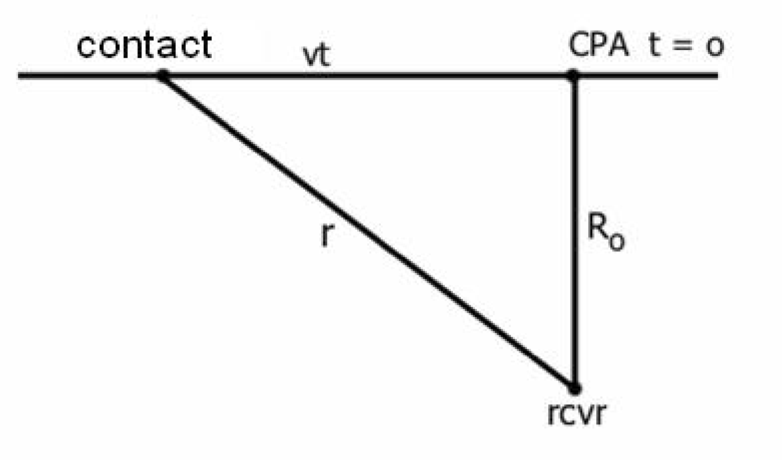


Figure 3. Geometry of contact passing through CPA. From [8].

The contact has a constant course and speed in this geometry where speed is given by *v*, range at CPA is given by R_o, and setting time at CPA $t=0$. This can then be arranged to give range to the contact as

$$r=\sqrt{R_o^{\ 2}+(vt)^2} \tag{0.0}$$

Substituting an expression for null frequencies into this can be rearranged to give

$$\mathbf{R}_{\mathbf{o}}^2=\left(\frac{2hd}{nc}\right)^2 f_n^2-v^2t^2. \tag{0.0}$$

As time goes to infinity, R_o becomes negligible compared to *vt* and the previous expression can be expressed as

$$t\cong\frac{2hd}{ncv}f_n. \tag{0.0}$$

This will then give a slope for the asymptote of the Lloyd's mirror pattern. On the other hand, at time $t=0$, the frequencies which are nulled at CPA will be given by

$$F_n=\frac{nc}{2hd}\mathbf{R}_{\mathbf{o}}. \tag{0.0}$$

When a narrowband tonal is also observed, more information can be obtained from the Doppler shift in the signal. This can be used to obtain the center frequency

$$f_o = \frac{f_u + f_l}{2}, \tag{0.0}$$

where f_u is the higher frequency and f_l is the lower frequency. The Doppler shift can be expressed as

$$\Delta f = f_u - f_o = \frac{f_o v}{c}. \tag{0.0}$$

This can then be arranged to give the speed of the contact, v

$$v = \frac{c\Delta f}{f_o} = \frac{c(f_u - f_l)}{f_u + f_l}. \tag{0.0}$$

Using these equations on a Lloyd's mirror pattern it is possible to obtain information on a contact's speed, depth and range at CPA [8].

Taking the derivative of the equation valid at long range where $r \cong vt$ with respect to time yields,

$$\frac{df_n}{dt} \cong \frac{ncv}{2hd}. \tag{0.0}$$

Dividing both sides by v gives

$$\frac{df_n}{v\,dt} \cong \frac{nc}{2hd}. \tag{0.0}$$

Since at long range $dr \cong vdt$, using (3.12) and (3.21) leads to the relationship

$$\frac{df_n}{dr} \cong \frac{f_n}{r}. \tag{0.0}$$

C. SHORT RANGE RAY THEORY ANALYSIS OF LLOYD’S MIRROR PATTERNS

For short ranges, the approximation that $r >> h + d$ is no longer valid, and therefore the simplified expression for the path length difference given by (3.11) cannot be used. On the other hand, the acoustic paths are more likely to be well-approximated as straight lines. For interference resulting from direct and surface reflected waves, the nulls will still be expected when the path length difference equals an integral number of wavelengths as before. Following the reasoning in [33] the path length of the surface reflected path, r_s, is given by

$$r_s = \sqrt{r^2 + \left(h + d\right)^2}\,, \tag{0.0}$$

and the direct path length, r_d, is given by

$$r_s = \sqrt{r^2 + \left(h - d\right)^2}\,. \tag{0.0}$$

So the condition for obtaining a null in the interference pattern becomes

$$\Delta r = r_s - r_d = \sqrt{r^2 + \left(h + d\right)^2} - \sqrt{r^2 + \left(h - d\right)^2} = n\lambda\,. \tag{0.0}$$

This expression cannot be simplified to the extent that the long range solution can. The depth of the receiver used in this work is too large. The largest path length differences will be obtained when the contact goes directly overhead. This scenario will result in the lowest possible nulled frequencies. In this case, the path length difference is simply $2d$ since the surface reflected path has to go from the source depth, d, to the surface and then back past the source down to the receiver. This implies that for a contact which goes by directly overhead, the lowest nulled frequency is determined by

$$\Delta r = 2d = n\lambda = \frac{nc}{f_n} \quad \text{or} \quad f_n = \frac{nc}{2d}\,. \tag{0.0}$$

As an example, a vessel with an effective draft of 10m in terms of its radiated noise would have a lowest nulled frequency of about 75Hz. As it moves away from overhead to larger horizontal ranges, the nulled frequencies will increase.

D. MODE THEORY

Mode theory is an alternative method of understanding the propagation of acoustic energy. It treats the ocean as a waveguide and breaks down propagating waves into a set of normal modes. Each of these normal modes represents an independent solution to the wave equation. The general form of a range independent normal mode solution is given by [8]

$$\hat{p}_t(r,z,t) = -j\pi e^{j\omega t}\sum_n Z_n(z_o)Z_n(z)H_o^{(2)}(k_n r) \quad . \qquad (0.0)$$

For large r this can be approximated as

$$\hat{p}_t(r,z,t) = -j\sum_n \sqrt{\frac{2\pi}{k_n r}} Z_n(z_o)Z_n(z)e^{j(\omega t - k_n r + \frac{\pi}{4})} \quad . \qquad (0.0)$$

In the ocean, where the sound speed profile is not constant and the boundary conditions not simple, the functions describing the dependence of the sound pressure on depth and the wavenumbers cannot normally be obtained analytically. Computer models must be used to obtain estimates of transmission loss for realistic conditions. However, (3.28) shows some of the important features of mode theory. It shows how the total pressure is the sum of all excited modes, that the spreading is expected to be cylindrical at long range, and that the phase of the modes at a given range will be different due to the modal dependence of the wavenumber, k_n.

E. WAVEGUIDE INVARIANT

An alternative way of predicting the Lloyd’s mirror pattern is the waveguide invariant approach. This concept assumes that the sound field consists of a limited number of closely spaced modes. It is generally valid at long ranges beyond where the simple two path Lloyd’s mirror pattern is valid. At long ranges, the higher order modes which interact more frequently with the boundaries will have been stripped off leaving the lower order modes. The waveguide invariant, β, is defined as

$$\beta = \frac{r}{\omega}\frac{d\omega}{dr} = -\frac{d\left(\frac{1}{v}\right)}{d\left(\frac{1}{u}\right)}, \qquad (0.0)$$

where u is the group speed and v is the phase speed of the modes. When the range between the source and receiver is much greater than the range at CPA, the range is approximately equal to vt and thus $dr \cong vdt$. Thus the waveguide invariant can be expressed in terms of time as

$$\beta = \frac{r}{\omega}\frac{d\omega}{dr} = \frac{r}{\omega}\frac{d\omega}{vdt}. \qquad (0.0)$$

This expression can be written in terms of frequency instead of angular frequency and rearranged as

$$\frac{df}{dr} = \beta\frac{f}{r} \text{ or } \frac{df}{dt} = \beta\frac{vf}{r}. \qquad (0.0)$$

In this form, the slope of the striations on a lofargram are given explicitly. When β is equal to 1, this waveguide invariant solution is identical to the long range ray theory solution given by (3.22) as pointed out in [14].

The waveguide invariant can have other values as well. Brekhovskikh and Lysanov [22] give an example of the waveguide invariant for a surface sound channel. For frequencies well above the cut-off frequency, $\beta \cong -3$. This value for the waveguide invariant would result in a dramatically different appearance for the striations. Taking (3.31) and integrating yields

$$\int\frac{df}{f} = \beta\int\frac{dr}{r}, \text{ or } \qquad \ln\frac{f}{f_0} = \beta\ln\frac{r}{r_0}. \qquad (3.32)$$

Exponentiating both sides yields the relationship

$$f = f_0 \left(\frac{r}{r_0} \right)^{\beta} \qquad (3.33)$$

This expression predicts how the frequency of a striation changes with range.

F. SUMMARY OF LLOYD’S MIRROR THEORY

The propagation of acoustic energy in the ocean is complex to model, and both ray theory and mode theory rely on assumptions which limit the range of their validity. At very short ranges where the horizontal range is smaller or only slightly larger than the combined depth of source and receiver, straight line ray theory with an exact solution of the path length difference is expected to provide the best estimate of the null frequencies as a function of range. If the range is much larger than the source and receiver depth (but not large enough to allow for additional significant ray paths between source and receiver), an approximation to the path length difference can be made which greatly simplifies the calculation. In fact, using this approximation one can determine velocity, range, and depth of a crossing contact from its lofargram if a Lloyd’s mirror pattern is displayed. An additional potential issue for ray theory is that it can have trouble at low frequencies where diffraction and other wave properties may manifest themselves more strongly.

In general, mode theory is used at longer ranges and lower frequencies. Even when these conditions are met, analytic solutions to mode theory are only tractable for simplified ocean conditions. Computer models need to be run for realistic conditions. On the other hand, the waveguide invariant concept, derived from mode theory, gives some insight into how the striations should change with range for various values of the invariant. Thus it is a valuable addition to understanding interference patterns in ocean acoustics.

IV. DATA ANALYSIS AND RESULTS

A. MATLAB

The first step in this thesis was to obtain the data from the OBS sensors and convert it into a usable format. After communicating with Doug Toomey, who controls the access to the data from this sensor network, permission was obtained to access and download the data. With the passwords provided, the data for the OBS network was then downloaded from the IRIS Data Management Center (DMC). These files were in a MiniSEED (mseed) file format, which is commonly used in seismology. At this point the mseed data was then converted to Seismic Analysis Code (SAC) for easier processing using MATLAB. The conversion to the SAC formatting was done by a program called mseed2sac. With the OBS data now in the SAC format, the data needed to be split into smaller chunks of data in order for the computer to be able to process it. Using MATLAB, the OBS data was split into the channels of interest, which consisted of the three seismometer channels and the one hydrophone channel. These four channels were then further split into each day from the start of the collection. This data, now in increments of one day, were then able to be processed. A MATLAB program was then run on the OBS data to produce a lofargram for evaluation as seen in Figure 4.

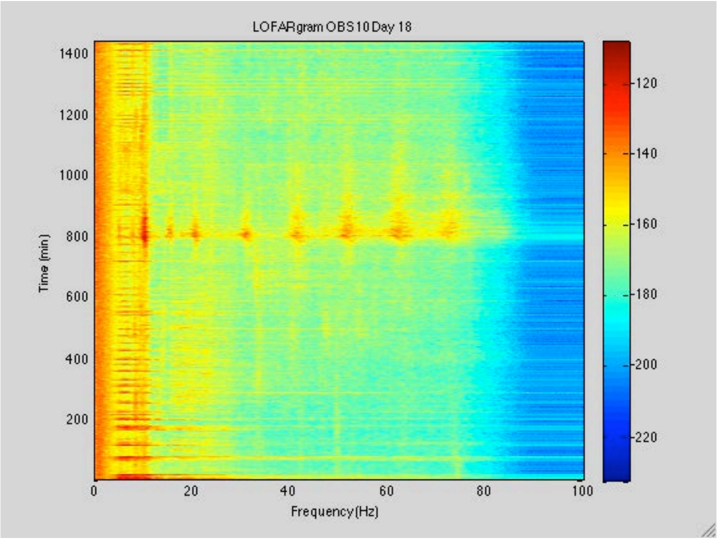


Figure 4. Lofargram for OBS10 on day 18

While evaluating these lofargrams, there were several contacts detected by the OBS sensors, and these times were noted for further evaluation. Each of these signals that were examined had a high SNR, a difference of about 40 dB, to ensure that there would be a strong enough signal to overcome the ambient noise in the area. A closer look at some of the contact signals shows a Lloyd's mirror pattern.

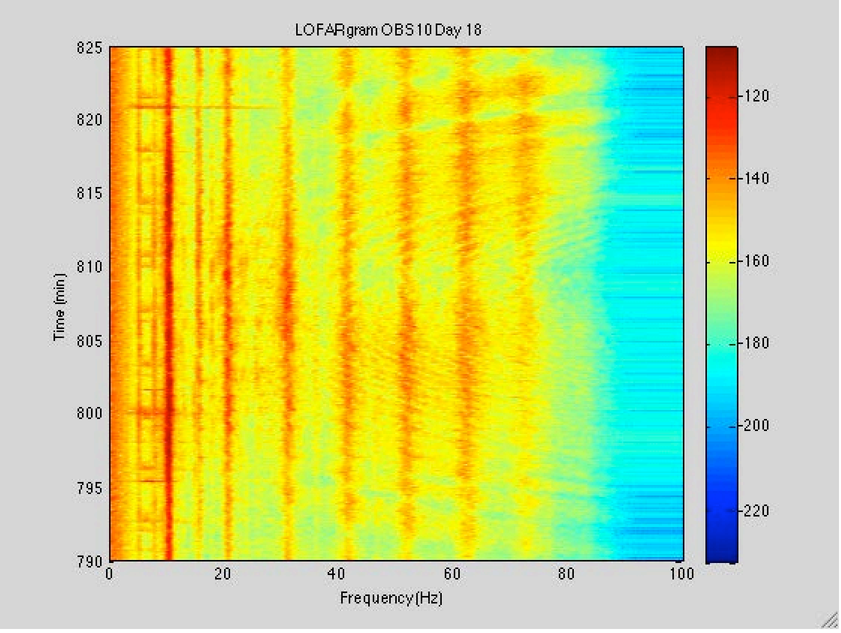


Figure 5. Example of Lloyd's mirror pattern

As seen in Figure 5, there is a clear Lloyd's mirror pattern in the lofargram. Lloyd's mirror patterns were seen in several of the contact signals, both on different sensors and different days.

B. BEAMFORMING

The seismometers were each oriented in a different direction: one north-south, one east-west, and one vertically. The coordinate system used to analyze the beamformer output is shown below in Figure 6. The north-south sensor is oriented along the y-axis. The east-west sensor is oriented along the x-axis, and the vertical sensor is positive in the z-direction.

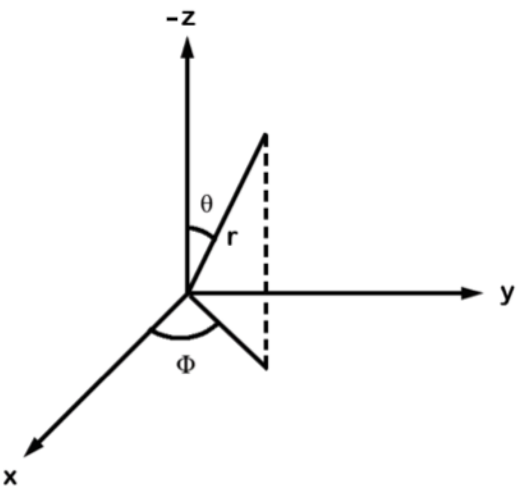


Figure 6. Coordinate system definition for beamforming

The seismometer and hydrophone data was then compiled and run through a MATLAB program to beamform the data. The goal was to see if this could be used to track a passing merchant as it was passing through the area of the OBS sensor. As seen in Figure 7b, the data appears to show the contact passing directly overhead since the strongest beamformer output occurs for a polar angle of 180^0.

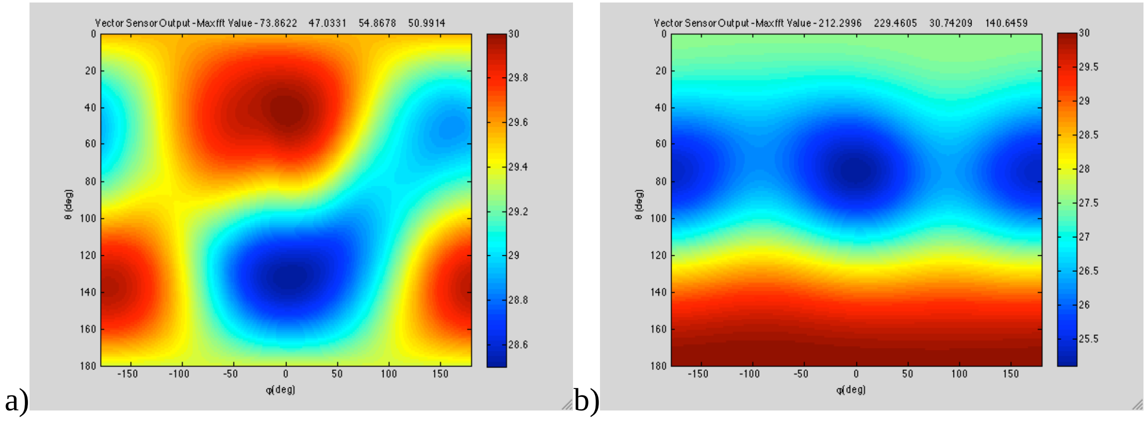


Figure 7. Beamformer output of OBS10 data, day 18, 809(a)-810(b) min

While this appears to give a bearing to the contact it was discovered that this bearing, over time, did not track as expected for a passing vessel. Instead, the bearing would jump erratically between several positions when it should be a smooth progression from one bearing to another as the vessel transited the area. Upon further investigation it was discovered that the phases of the data on each of the seismometers were very erratic. As shown in the theory, a single path acoustic signal will be either in phase or 180^0 out of phase with the pressure. If it is not in phase the data will interfere with each other when added causing the beamformer output to be degraded. This erratic phase difference can be seen in Figure 8. The phase of the pressure is compared to the phase of each of the seismometer channels, and the phase between two of the seismometers, east and north, are compared as well. The phase differences are never steady.

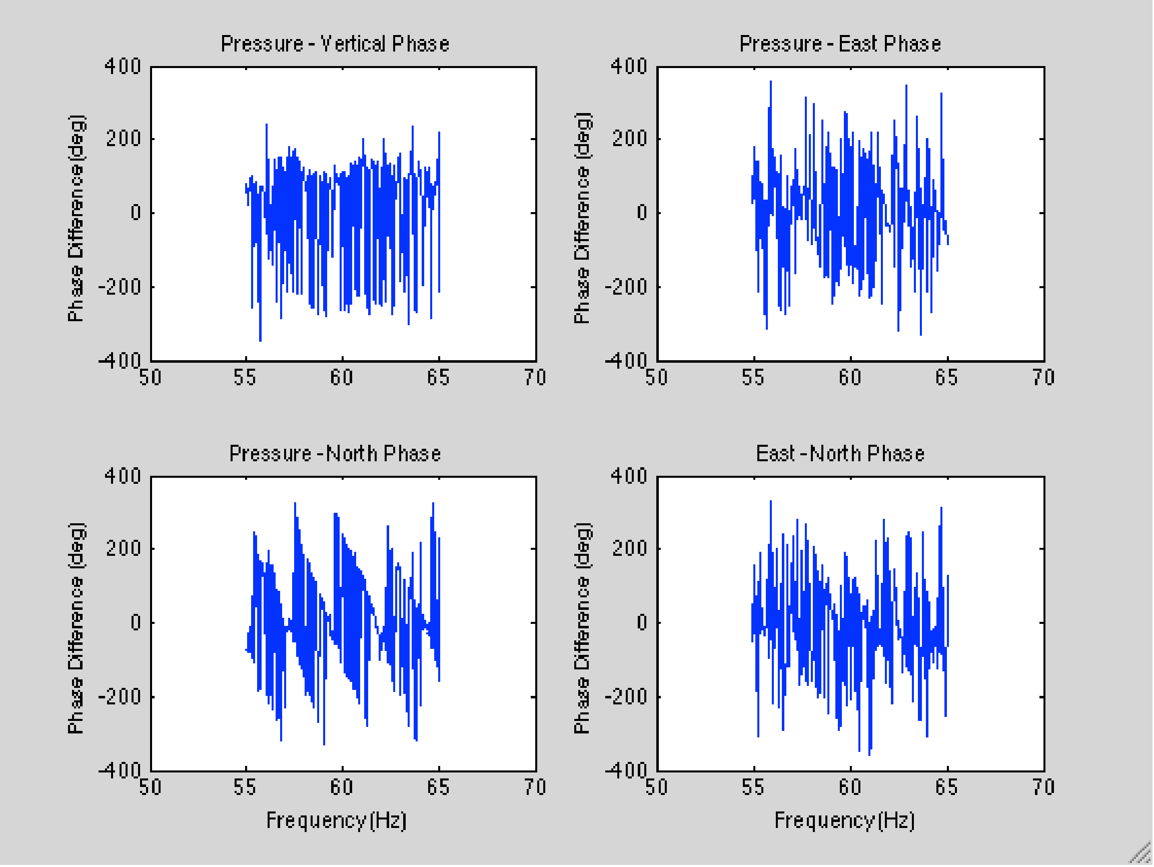


Figure 8. Example of phase difference between the seismometers (east, north, and vertical), and hydrophone (pressure)

There is also a lack of coherence between the signals as seen in Figure 9.

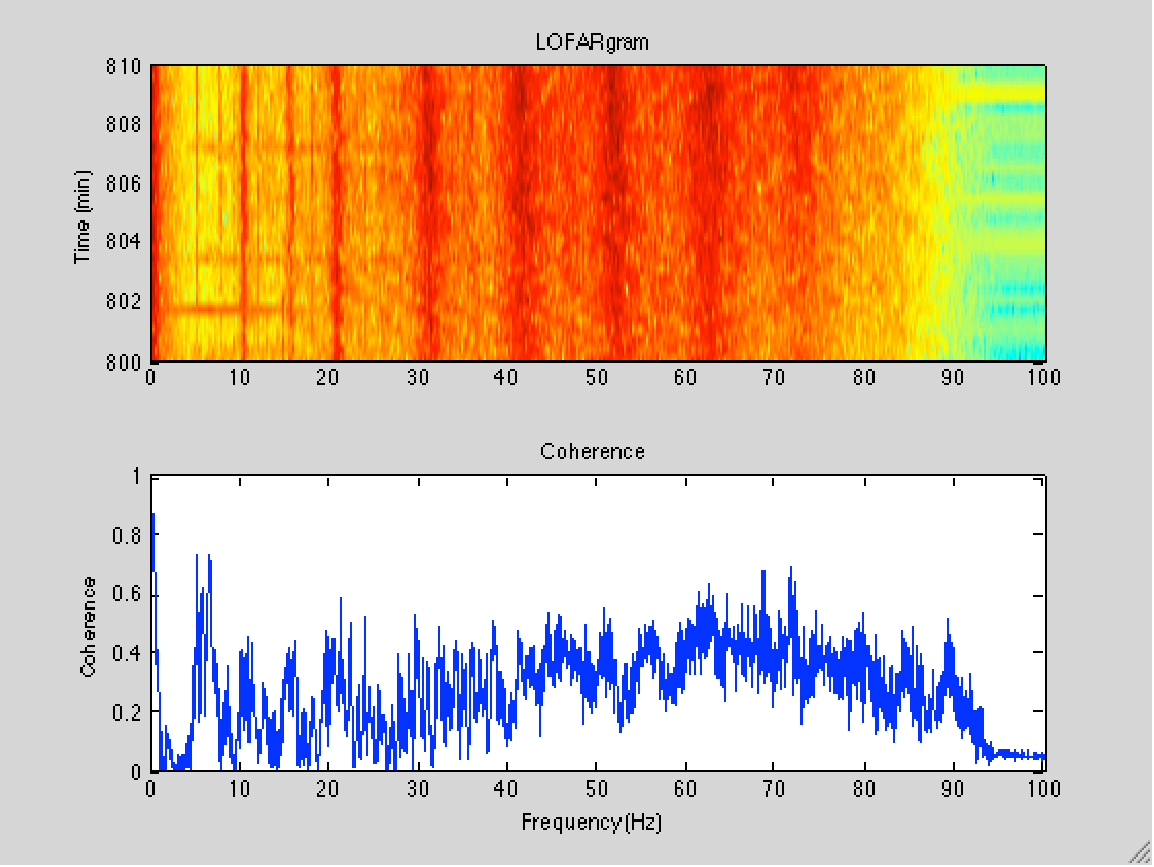


Figure 9. Example of coherence between east and north seismometers

As discussed in [14], [15], [16], [17] the erratic phase and lack of coherence are most likely the result of the acoustic energy coupling into surface waves along the ocean bottom as well as other waterborne multipath effects.

To check if these issues with phase differences and lack of coherence were limited to one OBS, several other signals from different OBSs were also examined. As an example, results are shown in Figures 10–13 below from OBS05. This sensor showed a weak Lloyd’s mirror pattern on day 21 at about 1325min. The beamformer output fails to display a smooth bearing change with time, and the phase difference between pressure and velocity channels is also erratic as seen before. There appears to be a range of frequencies for which the phase difference between the east-north seismometer channels look reasonable. However, it is not clear that this information could be trusted to determine anything about the azimuthal bearing due to the complexity of sound coupling into the bottom as well as possible extraneous seismic signals.

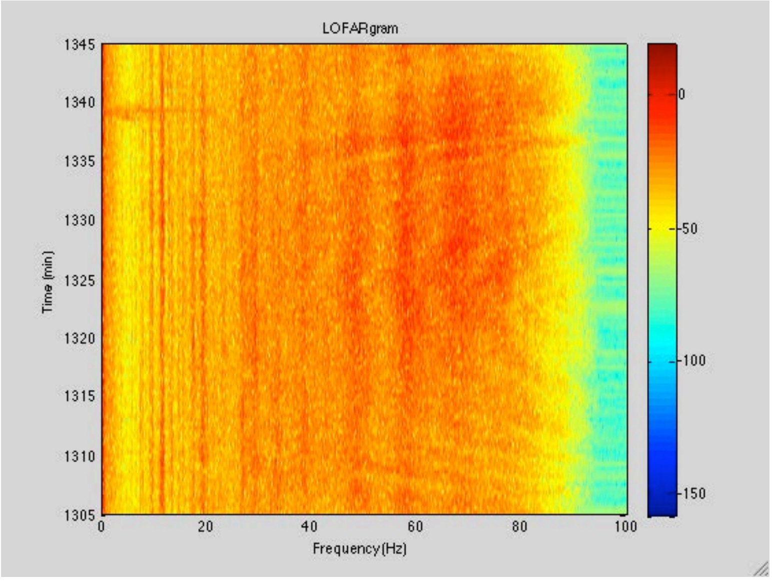


Figure 10. Lofargram for OBS05 on day 21

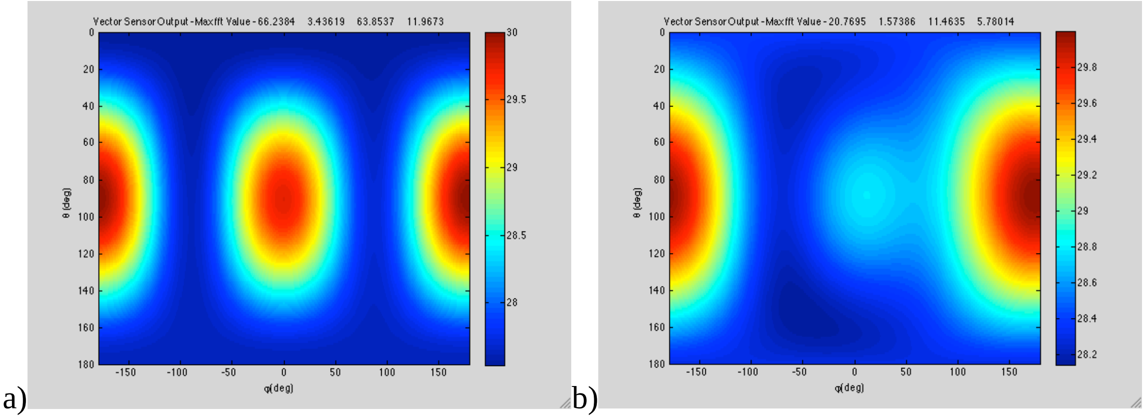


Figure 11. Beamformer output of OBS05 data, day 21, 1325(a)-1326(b) min

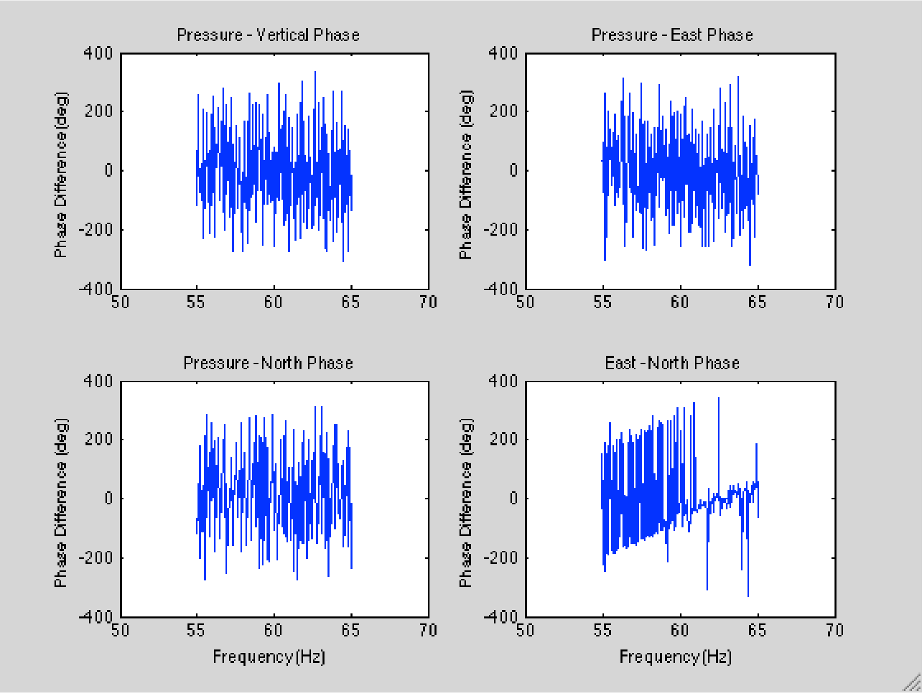


Figure 12. Example of phase difference between the seismometers (east, north, and vertical), and hydrophone (pressure)

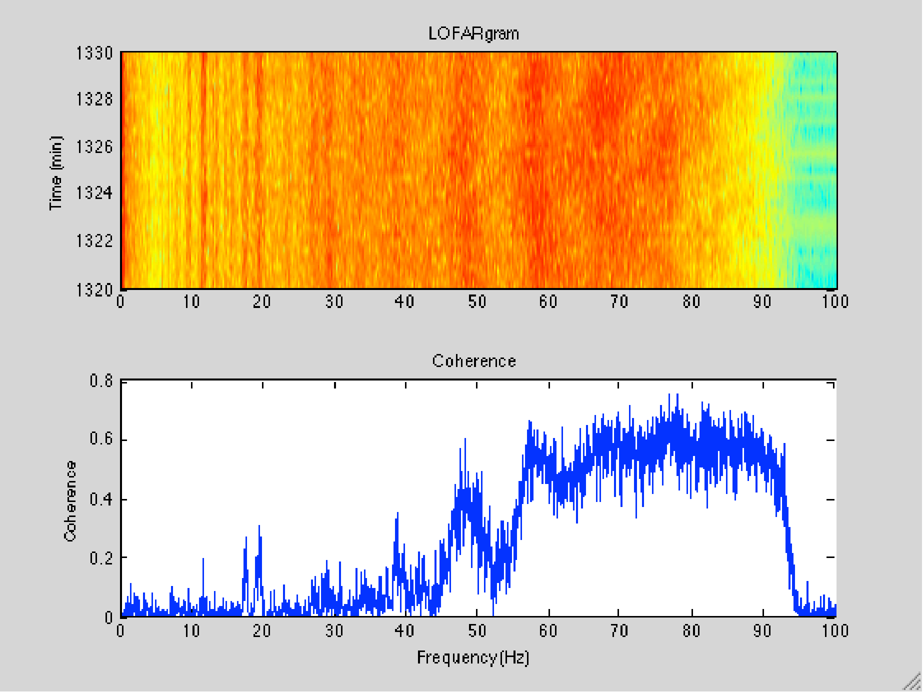


Figure 13. Example of coherence between east and north seismometers

C. LLOYD’S MIRROR

The existence of Lloyd’s mirror patterns in the OBS data suggests the intriguing possibility of exploiting them to derive valuable information about a contact. Hudson [20] showed that a combination of the Doppler shift and Lloyd’s mirror pattern could be used to determine the velocity, range, and depth of a contact. D`Spain and Kuperman [23], Thode [24], and Rouseff and Leigh [27] suggest that long range interference patterns can be exploited for range and possibly depth.

Unfortunately, the simple long range ray theory method is unlikely to work for the OBS data used in this study. Since the sensors are located on the ocean bottom, the source would have to be at least several kilometers from the receiver to satisfy the criteria that the horizontal range be much larger than the combined receiver and source depth. At that distance, ray curvature and multipath effects would need to be taken into account.

On the other hand, short range ray theory should yield reasonable predictions for the position and evolution of the Lloyd’s mirror pattern. The velocity of a contact could either be estimated from the Doppler shift or based on typical ship speeds. The Doppler estimation is a bit trickier at such low frequencies. Since it is smaller, a longer period of time is required to measure it accurately.

MATLAB was used to investigate the prediction of the short range ray theory. A contact was assumed to be transiting at a speed of 15kts with different CPA ranges and depths. The results are shown in Figure 14 below. Differences in the observed frequency, spacing, and slope of the striations result from the different ranges and depths. As the depth increases, the lowest frequency observed in the Lloyd’s mirror pattern decreases as well. Larger CPAs result in larger spacing of the striations as well as a shift to higher frequencies.

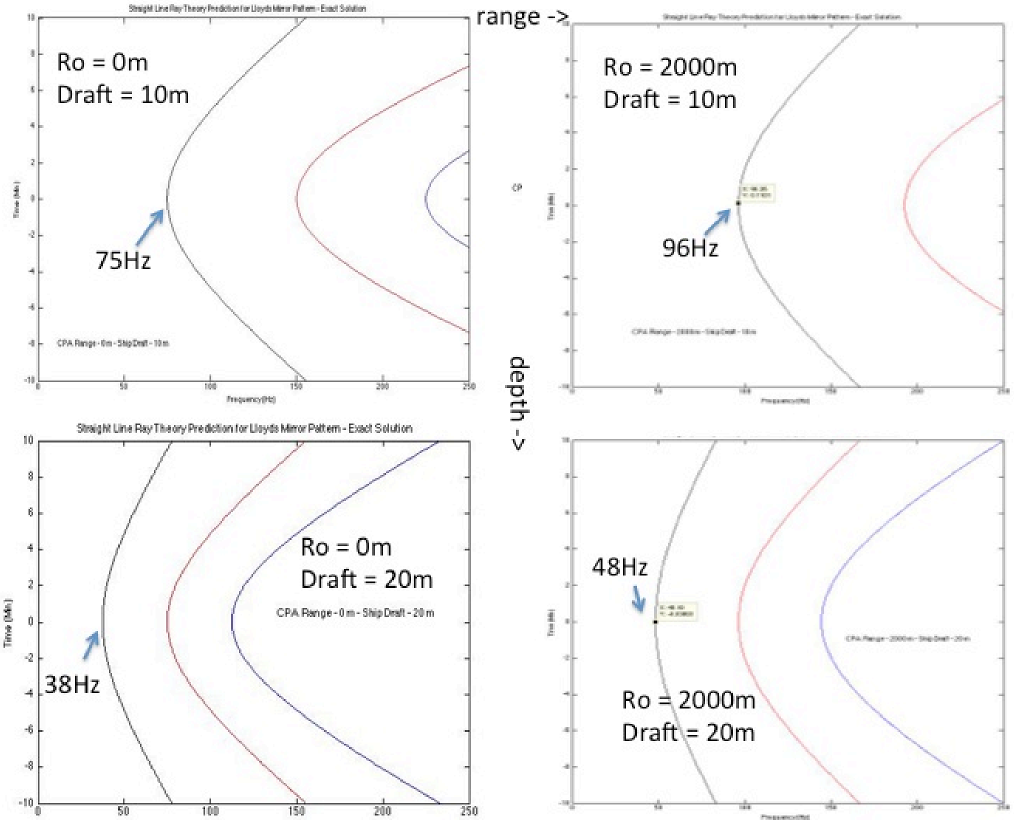


Figure 14. Nulled frequencies predicted by short range ray theory as a function of CPA range and depth

At longer ranges, ray theory breaks down and mode theory needs to be used. The waveguide invariant can shed some light on the observed interference patterns. If the waveguide invariant is known, the slope of the striations with respect to range can be predicted at a point on the frequency-range plot. However, the waveguide invariant itself cannot be used to predict the exact location of the nulls. Besides which, the waveguide invariant depends on the bathymetry and sound speed profile and would be hard to determine even in non-range dependent environments. Therefore, computer modeling was chosen to investigate the long range behavior of the expected interference patterns.

Because the data being analyzed was very low frequency (0–80Hz), the Parabolic Equation (PE) model in PC-IMAT was used to generate transmission loss data for the specific latitude and longitude of OBS10. The date at which the data was collected was also entered into PC-IMAT in order to get a realistic sound speed profile from its historic database. The database models a Sea State 3 with a total ambient noise of 67dB. The input fields for the PE model also include the frequency at which the transmission loss (TL) is to be calculated. However, discussions with the developers revealed that a range of frequencies can be specified. PC-IMAT takes the frequency range and computes a separate matrix of transmission loss as a function of range and depth at discrete frequencies within this bandwidth. The user cannot specify exactly which frequencies within the bandwidth are used, but they are spaced reasonably closely. For the 5–80Hz bandwidth, the frequencies at which the TL was calculated are shown in Figure 15.

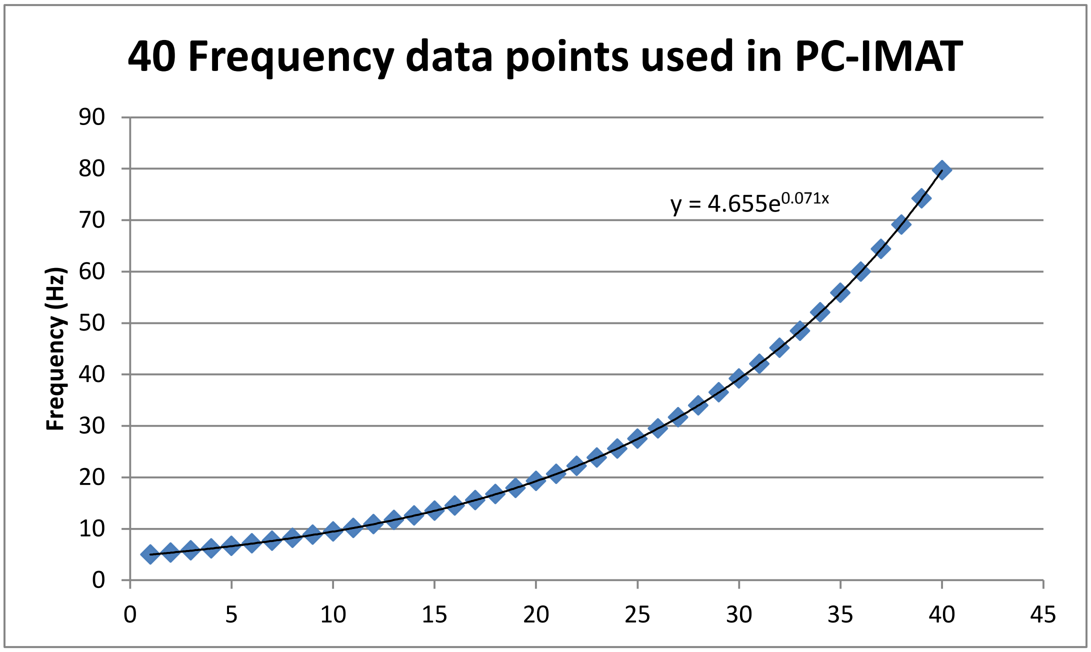


Figure 15. Frequencies used by PC-IMAT for TL calculations

PC-IMAT also allows the user to select the direction in which the TL is to be calculated. For this work the TL was calculated due north of the sensor. Because the bathymetry is fairly flat in the area, the TL along other bearings should be very similar.

It should be noted at this point that the maximum transmission loss corresponds to the nulls in a lofargram, so the two figures are in some sense photographic negatives of each other. Another difference which is important to note is that the lofargram is plotted as a function of time, whereas TL is plotted as a function of range. For a crossing contact with constant course and speed, converting between the two requires an assumption of speed and CPA.

Taking a MATLAB program provided by the PC-IMAT developers [34], the raw data files for the transmission loss at each frequency were converted from PC-IMAT into a format usable by MATLAB. The resulting matrices of TL vs range and depth were calculated at the same values of the depth, but the ranges at which the TL was computed were not always consistent. To make the matrices for each frequency representative of the TL at the same values of depth and range, a two dimensional interpolation was performed in MATLAB and compared against each original plot. The transmission loss data was then combined into a three dimensional matrix in terms of range, depth, and frequency. To produce plots of transmission loss as a function of range and frequency, a depth was set and the associated range and frequency data was plotted. Figure 15 shows the Sound Speed Profile (SSP) used by PC-IMAT for these calculations. The sound speed at the surface is 4931ft/sec (1503m/s), at the sofar axis it is 4844ft/sec (1476m/s), and at the bottom is 4919ft/sec (1499m/s).

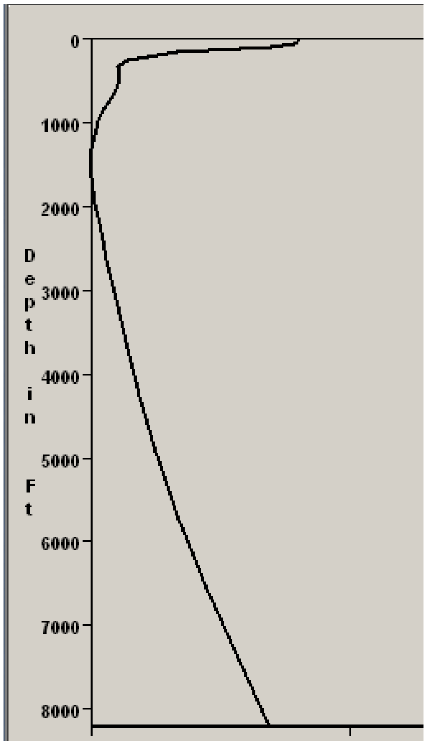


Figure 16. SSP for OBS 10

To investigate the effect of source depth and range on transmission loss, the source depth was set in ten foot increments from the surface down to 100 feet, and in 100 foot increments from 100 feet to the bottom. The TL data was then plotted as a range vs. frequency plot for analysis. An examination of these plots shows that there is no noticeable interference pattern in the 0–80Hz bandwidth for contacts less than 5,000 yards in range unless it also has a draft greater than about 50 feet, as seen in Figures 16–20.

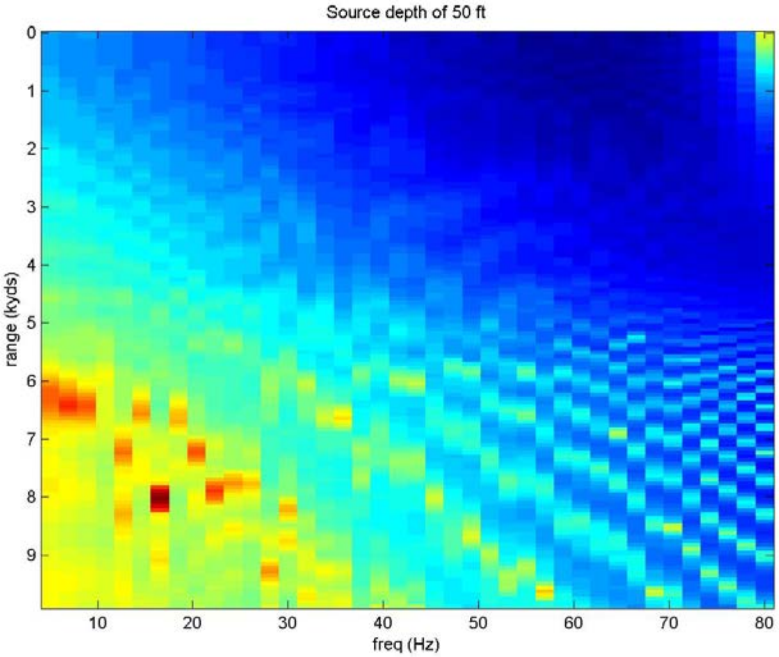


Figure 17. Source depth of 50 feet. Null starts to appear for contact directly overhead at about 80Hz.

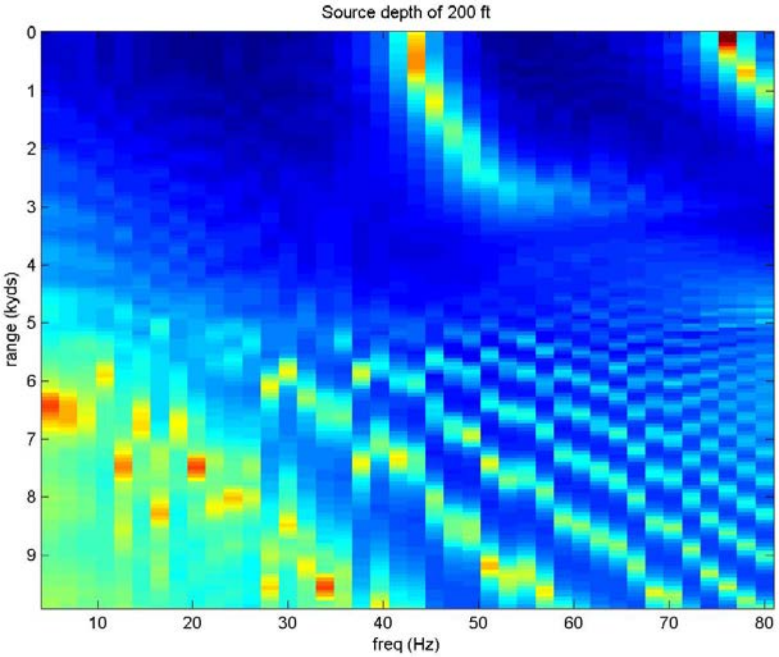


Figure 18. Source depth of 200 feet

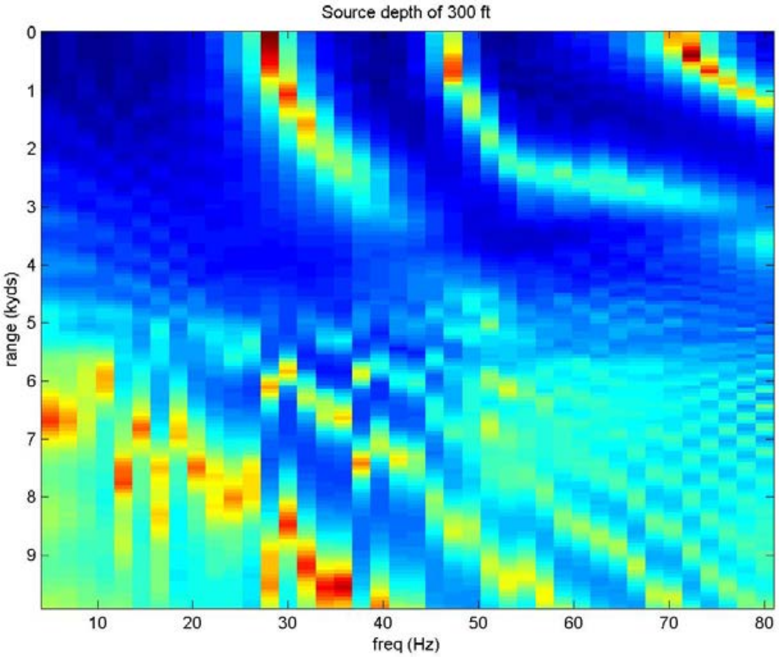


Figure 19. Source depth of 300 feet

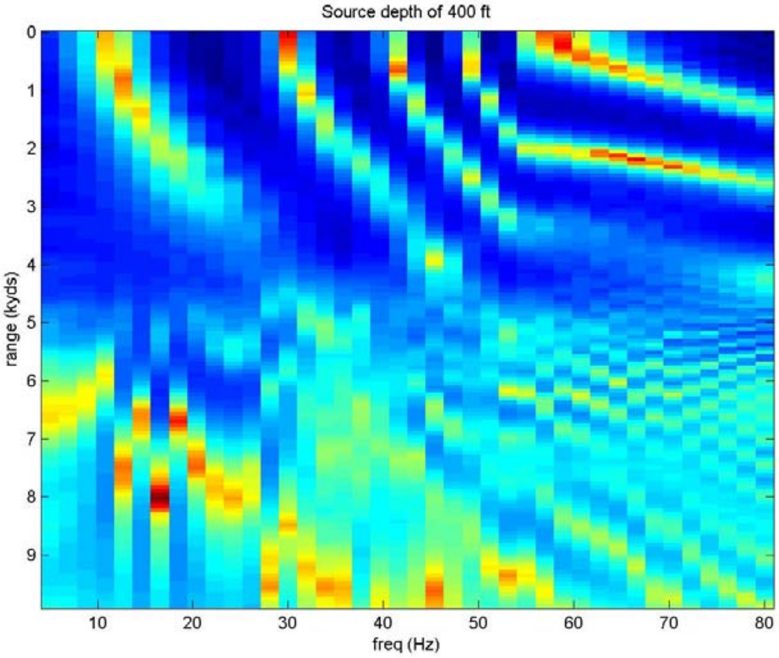


Figure 20. Source depth of 400 feet

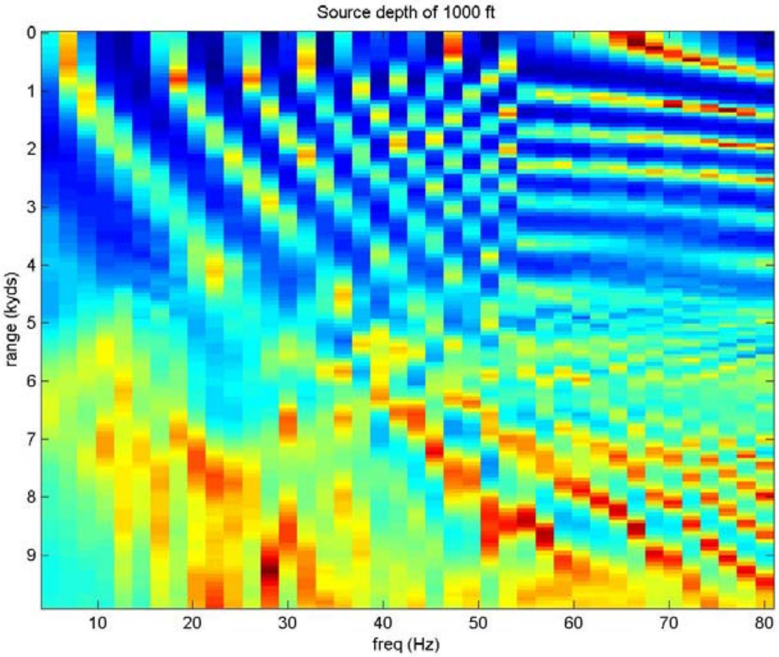


Figure 21. Source depth of 1000 feet

An interference pattern is noticeable for these deep contacts in the full range scale looked at (0 to 10,000 yards). In fact, as a close-in contact goes deeper, the nulls appear at lower and lower frequencies and are closer together.

Another noticeable result was that an interference pattern was detected for ranges greater than 5,000 yards on all contacts, regardless of the depth that was being looked at. These long range interference patterns look very similar qualitatively. So it is not clear whether they could be used to determine contact draft. On the other hand, the location of the nulls is clearly range dependent. These results are in agreement with previous work which suggests the use of the long range interference pattern to determine contact ranges.

The PC-IMAT developers clarified that the Parabolic Equation model is most effective at 20 degrees about the horizontal [35]. For close ranges where the contact would be outside this 20 degree window, the PE model uses a ray trace instead. This can explain the apparent transition at about 5,000 yards in the transmission loss plots.

Considering the fact that PC-IMAT uses ray theory for points where the PE model is invalid, it is surprising that PC-IMAT's predictions for the frequencies nulled at very close range are not exactly as predicted by straight line ray theory. Further work is needed to understand this discrepancy.

V. CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

The attempt to use the OBS sensors to determine bearings to a contact proved to be unsuccessful. Even with a high SNR of about 40 dB, the bearings to the contact were erratic. Using the hydrophone and three orthogonal seismometer channels as a vector sensor to determine contact bearing failed due to the lack of coherence and set phase difference between channels. This is most likely due to a variety of factors including the existence of multiple acoustic paths, coupling between the surface and waterborne acoustic waves, and various seismic signals in the area. As these combine and interfere with each other and the signal of interest, the amplitude and phase of the signal gets distorted preventing the directionality of the signal from being determined accurately.

Both straight line ray theory and models of transmission loss run in PC-IMAT show distinct differences in the interference patterns for sources at different depths at close range. This could be used as a way to determine the draft of contacts passing through a sensor field. The long range interference patterns predicted by PC-IMAT do not show an obvious difference as a function of depth.

B. RECOMMENDATIONS

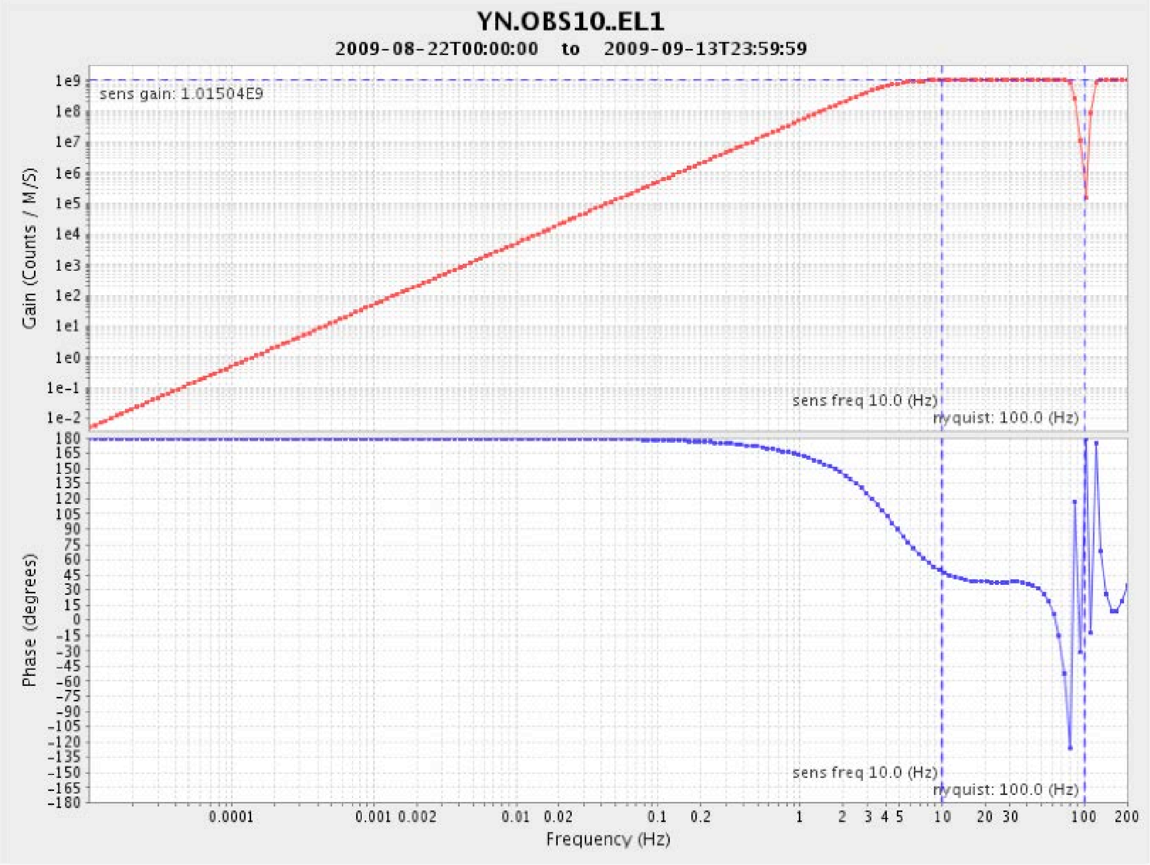
Further research into the interference patterns produced by PC-IMAT needs to be completed. In particular, the discrepancy between short range ray theory and PC-IMAT needs to be understood better—especially since the short range interference patterns appear to hold the most promise for range/draft determination. Even at long range, it may be possible to use interference as an aid in determining vessel draft and range. However, higher resolution modeling would be required. Further research should be conducted into matching high resolution interference pattern predictions assuming a variety of range, depth, and speeds with the lofargrams of contacts with known ground truth.

Another area to look into would be the use of multiple OBS sensors to beamform the data. This would depend on picking up the same contact simultaneously on multiple OBSs and having coherent signals between them. Although the sensors in this and other

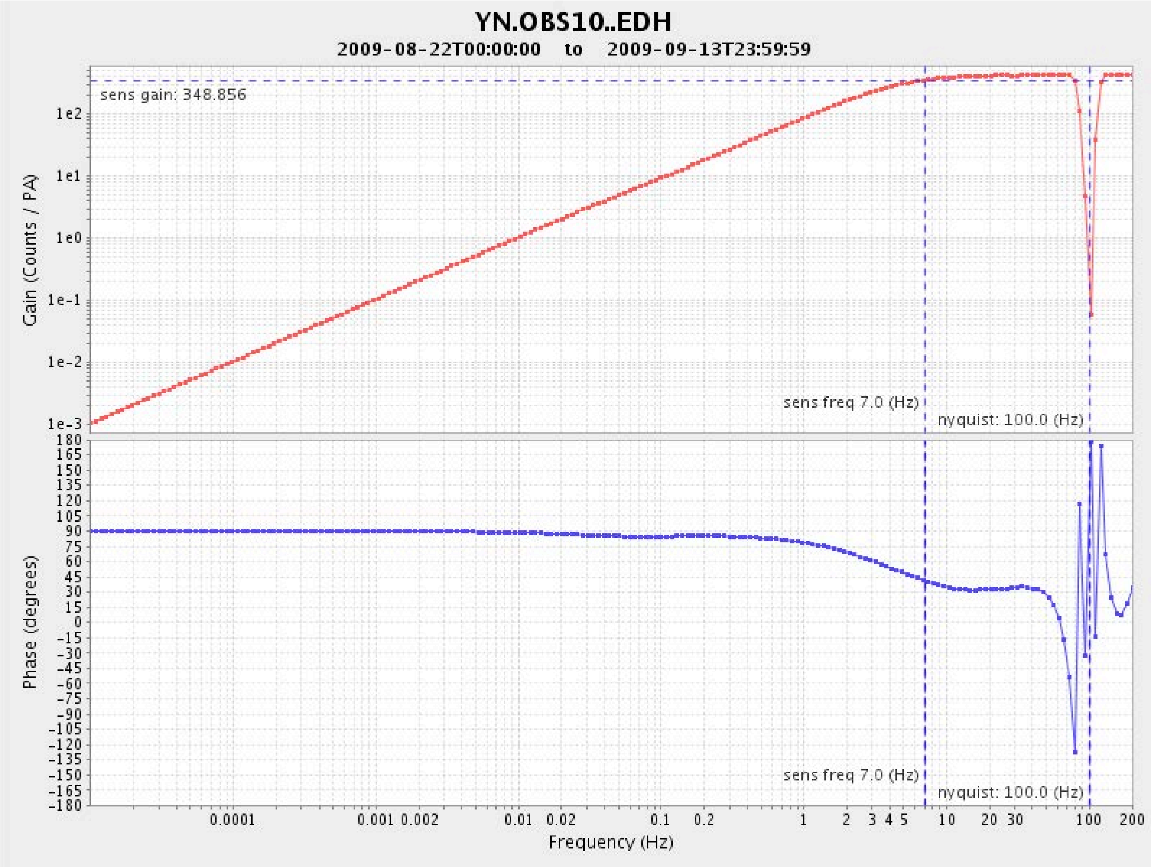
OOS networks are spaced fairly far apart, the low frequency signals suffer little in the way of absorption losses. With a network of sensors that could track contacts this could become a supplement to Automatic Identification System (AIS) and other tracking systems to help monitor shipping around the world.

APPENDIX A. CALIBRATION CURVES.

A. CALIBRATION CURVE FOR SIESMOMETERS



B. CALIBRATION CURVE FOR HYDROPHONE



APPENDIX B. MATLAB CODE.

A. OBS ANAYLYSIS

```
clear all

load OBS10_EDH_234_amp_18.mat
load OBS10_EL1_234_amp_18.mat
load OBS10_EL2_234_amp_18.mat
load OBS10_ELZ_234_amp_18.mat

sEDH=3.488560e+02;%sensitivity of hydrophone
sEL=1.015040e+09;%sensitivity of seismometers

p=OBS10_EDH_234_amp_18/sEDH;
u1=OBS10_EL1_234_amp_18/sEL;  % east
u2=OBS10_EL2_234_amp_18/sEL;  % north
uz=OBS10_ELZ_234_amp_18/sEL;  % vertical

rhoc = 1.54e6;
u1c = u1*rhoc*3;
u2c = u2*rhoc*3;
uzc = uz*rhoc*3;

% de-mean data

p = p - mean(p);
u1c = u1c - mean(u1c);
u2c = u2c - mean(u2c);
uzc = uzc - mean(uzc);

%
% factorH = 16;    % code to eliminate data factor times the standard
deviation
% factor1 = 16;
% factor2 = 16;
% factorz = 16;
%
% p_std = std(p);
%
% r = find(abs(p)>factorH*p_std);
%
% p(r) = sign(p(r))*factorH*p_std;
%
%
% u1c_std = std(u1c);
%
% r = find(abs(u1c)>factor1*u1c_std);
%
```

```
% u1c(r) = sign(u1c(r))*factor1*u1c_std;
%
% u2c_std = std(u2c);
%
% r = find(abs(u2c)>factor2*u2c_std);
%
% u2c(r) = sign(u2c(r))*factor2*u2c_std;
%
% uzc_std = std(uzc);
%
% r = find(abs(uzc)>factorz*uzc_std);
%
% uzc(r) = sign(uzc(r))*factorz*uzc_std;


bindata=[p uzc u1c u2c];  % matrix with full day/all 4 channels

% produce lofargram of desired channel

signal = p;   % allows user to choose which channel to produce
lofargram

Fs = 200;  % sampling frequency

NFFT = 2^12;             % length of FFT

window = hamming(NFFT);

avg = 1;                 % number of averages to take

chunk = NFFT*avg;     %   number of data points needed to get averages

N = floor(length(p)/chunk);  % Total number of chunks available

lofar = ones(N,NFFT/2+1);    % initialize the matrix to make for loop
faster

x = ones(1,NFFT/2+1);


for i = 1:N                  % calculate avg power spectrum for each
chunk

   low = 1+(i-1)*chunk;
   high = i*chunk;

[x,f] = pwelch(signal(low:high),window,NFFT/2,NFFT,Fs);

lofar(i,:)=x;     % make the ith row of the lofargram the power
```

```
spectrum for that chunk

t(i) = i*chunk/Fs;   % the time corresponding to the chunk

end

lofar = 10*log10(lofar);

t = t’;

t = t/60;  % time in minutes


figure(1)

pcolor(f,t,lofar)
shading interp
title(‘LOFARgram’)
xlabel(‘Frequency (Hz)’)
ylabel(‘Time (min)’)
colorbar

% look and listen to data

answer=2

while answer==1

    start_time = input(‘Input start time to look at data in minutes.’)
% start time for movie in minutes

    start_index = ceil(start_time*60*Fs)+1;

    end_time = input(‘Input end time for data in minutes.’)   % end
time for movie in minutes

    end_index = floor(end_time*60*Fs);

    signalt = signal(start_index:end_index);

    figure(2)   % look at signal in time domain

    time = (start_index:end_index)/(200*60);  % time in minutes

    plot(time,signalt)

    title([‘Time Domain Signal - Standard Deviation = ‘,
num2str(std(signalt))])
```

```
    xlabel(‘Time (min)’)

    s = input(‘Listen to signal? 1 for yes, 2 for no.’);

    if s == 1
    sound(signalt,Fs*10)  % listen to data at higher Fs % can hear
whole sequence by
                         % making wav wavwrite(signal,Fs*10,16,’test’)
    end

    answer = input(‘Do you want to look at another sequence? 1= yes 2 =
no’)

end



start_time = input(‘Input start time for movie in minutes.’)  % start
time for movie in minutes

start_index = floor(start_time*60*Fs);

end_time = input(‘Input end time for movie in minutes.’)   % end time
for movie in minutes

end_index = floor(end_time*60*Fs);

start_f = input(‘What frequency do you want to start with?’)

stop_f = input(‘What frequency do you want to end with?’)

avg = 1;  % not sure how to implement averaging in beamformer

binN = NFFT*avg; % data to take for each frame

chunks = ceil((end_time - start_time)*60*Fs/binN);

time_per_frame = binN/200  % time per frame of movie in seconds


for j=1:chunks

    data = bindata(start_index+(j-1)*binN:start_index+j*binN-1,:);

     % channel assignments
    prs = 1; % hydrophone
    ELZ = 2; % vertical seismometer (down or up? says “-90”)
    EL1 = 3; % east seismometer
    EL2 = 4; % north seismometer

    x = 1;   % define directions
    y = 2;
    z = 3;

    thetainc = 1; % angle increment degrees
    phiinc = 1; % angle increment degrees

    k = linspace(0,NFFT-1,NFFT); %baseline FFT bin numbers

    % vector sensor MRA angles in radians

    thetaELZ = 0; % vertical seismometer polar angle 0∞ (could be 180∞)
    phiELZ = 0; % vertical seismometer azimuthal angle 0∞ (doesn’t
really matter)
    thetaEL1 = pi/2; % eastern seismometer polar angle 90∞
    phiEL1 = 0; % eastern seismometer azimuthal angle 0∞ defining east
as +x-axis
    thetaEL2 = pi/2; % northern seismometer polar angle 90∞
    phiEL2 = pi/2; % northern seismometer 90∞ (defining north as +y-
axis)


    % sensor element unit vectors

    ue(prs,:) = [0 0 0]; % hydrophone [0 0 0] - change in udot to
[omni]
    ue(ELZ,:) =  [sin(thetaELZ)*cos(phiELZ) sin(thetaELZ)*sin(phiELZ)
cos(thetaELZ)]; % ELZ [0 0 1]
    ue(EL1,:) = -[sin(thetaEL1)*cos(phiEL1) sin(thetaEL1)*sin(phiEL1)
cos(thetaEL1)]; % EL1 [1 0 0]
    ue(EL2,:) = -[sin(thetaEL2)*cos(phiEL2) sin(thetaEL2)*sin(phiEL2)
cos(thetaEL2)]; % EL2 [0 1 0]

    % steered unit vectors

    thetas = 0:deg2rad(thetainc):pi; % polar steer angle array -
radians
    phis = -pi:deg2rad(phiinc):pi; % azimuthal steer angle - radians

    m = length(thetas); % total number of angles in theta direction
    n = length(phis); % total number of angles in psi direction

    u = sin(thetas’) * cos(phis); % steered direction cosine x - m x n
matrix
    v = sin(thetas’) * sin(phis); % steered direction cosine y - m x n
matrix
    w = cos(thetas’) * ones(1,n); % steered direction cosine z - m x n
```

```
matrix

    % element unit vector & steer angle dot product array
    % vectorize dot products for faster multiplication in beamformer

    udot = ue(prs:EL2,x)*u(:)’ + ue(prs:EL2,y)*v(:)’ +
ue(prs:EL2,z)*w(:)’; %1–4 x m*n

    % omni-directional pressure sensors - vector is “1” in all
orientations and aspects

    udot(1,:) = ones(1,m*n); %1 x m*n

    f = k*Fs/NFFT; % vector of frequencies corresponding with bin
numbers

    low_f = max(find(f<start_f));
    high_f = min(find(f>stop_f));
    fu = f(low_f:high_f);
    o = length(f(low_f:high_f));

    dataF = ones(NFFT,4);

   % hanning window to reduce sidelobes of signal

    H = hann(NFFT) * ones(1,4);

    size(H);

    size(data);

    data = data .* H;

    % convert windowed time domain signal to frequency domain

    dataF = fft(data); % by columns N x 4

    dataF = dataF(low_f:high_f,:);

    intensity_channels = mean(abs(dataF));

    dataF = dataF.’; % change to row data 4 x N, non conjugate
transpose

    dataF = conj(dataF);   % can’t remember why we did this

    figure(2)
    subplot(2,2,1)
    plot(fu,abs(dataF(1,:)))
    title(‘Pressure’)
    subplot(2,2,2)
     plot(fu,abs(dataF(2,:)))
    title(‘Corrected Vertical Seismometer’)
    subplot(2,2,3)
     plot(fu,abs(dataF(3,:)))
    title(‘Corrected East Seismometer’)
    xlabel(‘Frequency (Hz)’)
    subplot(2,2,4)
     plot(fu,abs(dataF(4,:)))
    title(‘Corrected North Seismometer’)
      xlabel(‘Frequency (Hz)’)
     figure(3)
    subplot(2,2,1)
    plot(fu,rad2deg(angle(dataF(1,:))))
    title(‘Pressure’); ylabel(‘Phase (deg)’)
    subplot(2,2,2)
     plot(fu,rad2deg(angle(dataF(2,:))))
    title(‘Corrected Vertical Seismometer’); ylabel(‘Phase (deg)’)
    subplot(2,2,3)
     plot(fu,rad2deg(angle(dataF(3,:))))
    title(‘Corrected East Seismometer’); ylabel(‘Phase (deg)’)
    xlabel(‘Frequency (Hz)’)
    subplot(2,2,4)
     plot(fu,rad2deg(angle(dataF(4,:))))
    title(‘Corrected North Seismometer’); ylabel(‘Phase (deg)’)
      xlabel(‘Frequency (Hz)’)
      figure(4)
      subplot(2,2,1)
    plot(fu,rad2deg(angle(dataF(1,:)) - angle(dataF(2,:))))
    title(‘Pressure - Vertical Phase’); ylabel(‘Phase Difference(deg)’)
    subplot(2,2,2)
     plot(fu,rad2deg(angle(dataF(1,:)) - angle(dataF(3,:))))
    title(‘Pressure - East Phase’);ylabel(‘Phase Difference (deg)’)
    subplot(2,2,3)
     plot(fu,rad2deg(angle(dataF(1,:)) - angle(dataF(4,:))))
    title(‘Pressure - North Phase’);ylabel(‘Phase Difference (deg)’)
    xlabel(‘Frequency (Hz)’)
    subplot(2,2,4)
     plot(fu,rad2deg(angle(dataF(2,:)) - angle(dataF(3,:))))
    title(‘East - North Phase’);ylabel(‘Phase Difference (deg)’)
      xlabel(‘Frequency (Hz)’)

    max_fft = max(max(abs(dataF)));

   %%initialize beamformer%% S is sum of all channels - the
beamformer%%

   S = zeros(m*n,size(dataF,2));  %final beamformer output - m*n x o
array

   %Beamformer output
```

```
    xcalc = zeros(m*n,o);
    udotc = zeros(1,m*n);
    sumcalc = zeros(m*n,o);

    for q = 1:4;
        %pull element matrix from udot array
        %Theta m * Phi n vector
        udotc = squeeze(udot((mod(q+3,4)+1),:,:)); %m*n x 1
        xcalc = udotc(:) * dataF(q,:); % m*n x o
        %Theta m x FFT o matrix
        %sumcalc - individual element output for each angle combo m*n x
p matrix
        %indexing alpha matrix to m*n x p faster than using “repmat”
with same effect
        sumcalc = xcalc;
        %S - to tal beamformer output for each angle combo m*n x p
matrix
        S = S + sumcalc;
    end

    %%Beamformer output%%
    %%Beamformer output%%

    S = sum(abs(S),2);  %sum along k “amplitude,” collapse to single
value in k direction

    S = reshape(S, [m n]); %mxn theta x phi 2D matrix

    S = S/max(max(S));

    SdB = 20*log10(abs(S)); %convert to dB

    SdB = SdB + 30;

    %remove SdB points outside 90% max value

    indices = find(SdB<0);

    SdB(indices) = 0;

    % angles in degrees

    phisdeg = rad2deg(phis);
    thetasdeg = rad2deg(thetas);


    %Overhead theta by phi amplitude plot
    figure (5)
    clims = [min(min(SdB)) max(max(SdB))];
    imagesc(phisdeg,thetasdeg,SdB,clims);
    title([‘Vector Sensor Output - Max fft Value - ‘, 
num2str(intensity_channels)])
    axis([-180 180 0 180])
    xlabel(‘\phi (deg)’)
    ylabel(‘\theta (deg)’)
    shading interp
    colorbar
    M(j)=getframe(2);
    pause



end
```

B. SONAR PROCESSOR TO PRODUCE LOFARGRAMS

```
%  Program to examine ocean OBS hydrophone data

% clf
% clear

load OBS10_EL1_234_amp_18.mat

y = OBS10_EL1_234_amp_18;

p = ‘OBS10 Day 234’;

sensitivity = 1.015040e+09;  % sensitivity of OBS10

y = y/sensitivity;  % convert to pressure in microPa

y = y - mean(y);

l = length(y);

Fs=200;

factor = 5;

y_std = std(y);                 % code to allow clipping data

r = find(abs(y)>factor*y_std);

y(r) = y_std;
```

```
NFFT = 2^9;             % length of FFT

window = hamming(NFFT);

avg = 2;                 % number of averages to take

chunk = NFFT*avg;     %   number of data points needed to get averages

N = floor(length(y)/chunk);  % Total number of chunks available

lofar = ones(N,NFFT/2+1);    % initialize the matrix to make for loop
faster
x = ones(1,NFFT/2+1);

for i = 1:N                  % calculate avg power spectrum for each
chunk

   low = 1+(i-1)*chunk;
   high = i*chunk;

[x,f] = pwelch(y(low:high),window,NFFT/2, NFFT,Fs);

x = x’;

lofar(i,:)=x;     % make the ith row of the lofargram the power
spectrum for that chunk

t(i) = i*chunk/Fs;   % the time corresponding to the chunk

end

lofar = 10*log10(lofar);

t = t’;

t = t/60;  % time in minutes


figure(1)

pcolor(f,t,lofar)
shading interp
title(‘LOFARgram OBS10 Day 18’)
xlabel(‘Frequency (Hz)’)
ylabel(‘Time (min)’)
colorbar
```

```
%Define the colormap if you want a grayscale

%c = linspace(0,1,256);
%caxis(‘auto’)
%C = [c;c;c]’;  % Colormap must have three columns. For grayscale each
column must have same value. This colormap has 256 grey tones.
%colormap(C)  % sets C as the colormap
```

### C.	RSAC

```
%RSAC    Read SAC binary files.
%    RSAC(‘sacfile’) reads in a SAC (seismic analysis code) binary
%    format file into a 3-column vector.
%    Column 1 contains time values.
%    Column 2 contains amplitude values.
%    Column 3 contains all SAC header information.
%    Default byte order is big-endian. M-file can be set to default
%    little-endian byte order.
%
%    usage:  output = rsac(‘sacfile’)
%
%    Examples:
%
%    KATH = rsac(‘KATH.R’);
%    plot(KATH(:,1),KATH(:,2))
%
%    [SQRL, AAK] = rsac(‘SQRL.R’,’AAK.R’);
%
%    by Michael Thorne (4/2004)   mthorne@asu.edu

function [varargout] = rsac(varargin);

for nrecs = 1:nargin

  sacfile = varargin{nrecs};

%----------------------------------------------------------------------
-----
%    Default byte-order
%    endian  = ‘big-endian’ byte order (e.g., UNIX)
%            = ‘little-endian’ byte order (e.g., LINUX)

endian = ‘little-endian’;

if strcmp(endian,’big-endian’)
  fid = fopen(sacfile,’r’,’ieee-be’);
elseif strcmp(endian,’little-endian’)
  fid = fopen(sacfile,’r’,’ieee-le’);
end
```

```
% read in single precision real header variables:
%----------------------------------------------------------------------
-----
for i=1:70
  h(i) = fread(fid,1,’single’);
end

% read in single precision integer header variables:
%----------------------------------------------------------------------
-----
for i=71:105
  h(i) = fread(fid,1,’int32’);
end



% Check header version = 6 and issue warning
%----------------------------------------------------------------------
-----
% If the header version is not NVHDR == 6 then the sacfile is likely of
the
% opposite byte order. This will give h(77) some ridiculously large
% number. NVHDR can also be 4 or 5. In this case it is an old SAC file
% and rsac cannot read this file in. To correct, read the SAC file into
% the newest verson of SAC and w over.
%
if (h(77) == 4 | h(77) == 5)
    message = strcat(‘NVHDR = 4 or 5. File: “‘,sacfile,’” may be from
an old version of SAC.’);
    error(message)
elseif h(77) ~= 6
    message = strcat(‘Current rsac byte order: “‘,endian,’.” File:
“‘,sacfile,’” may be of opposite byte-order.’);
    error(message)
end

% read in logical header variables
%----------------------------------------------------------------------
-----
for i=106:110
  h(i) = fread(fid,1,’int32’);
end

% read in character header variables
%----------------------------------------------------------------------
-----
for i=111:302
  h(i) = (fread(fid,1,’char’))’;
end

% read in amplitudes
```

```
%----------------------------------------------------------------------
-----

YARRAY     = fread(fid,’single’);

if h(106) == 1
  XARRAY = (linspace(h(6),h(7),h(80)))’;
else
  error(‘LEVEN must = 1; SAC file not evenly spaced’)
end

% add header signature for testing files for SAC format
%----------------------------------------------------------------------
-----
h(303) = 77;
h(304) = 73;
h(305) = 75;
h(306) = 69;

% arrange output files
%----------------------------------------------------------------------
-----
OUTPUT(:,1) = XARRAY;
OUTPUT(:,2) = YARRAY;
OUTPUT(1:306,3) = h(1:306)’;

%pad xarray and yarray with NaN if smaller than header field
if h(80) < 306
  OUTPUT((h(80)+1):306,1) = NaN;
  OUTPUT((h(80)+1):306,2) = NaN;
end

fclose(fid);

varargout{nrecs} = OUTPUT;

end
```

D. COHERENCE CHECK

```
clear all

load OBS10_EDH_234_amp_18.mat
load OBS10_EL1_234_amp_18.mat
load OBS10_EL2_234_amp_18.mat
load OBS10_ELZ_234_amp_18.mat

sEDH=3.488560e+02;%sensitivity of hydrophone
```

```
sEL=1.015040e+09;%sensitivity of seismometers

p=OBS10_EDH_234_amp_18/sEDH;
u1=OBS10_EL1_234_amp_18/sEL;  % east
u2=OBS10_EL2_234_amp_18/sEL;  % north
uz=OBS10_ELZ_234_amp_18/sEL;  % vertical

rhoc = 1.54e6;
u1c = u1*rhoc*3;
u2c = u2*rhoc*3;
uzc = uz*rhoc*3;

% de-mean data

p = p - mean(p);
u1c = u1c - mean(u1c);
u2c = u2c - mean(u2c);
uzc = uzc - mean(uzc);



%
% factorH = 16;    % code to eliminate data factor times the standard
deviation
% factor1 = 16;
% factor2 = 16;
% factorz = 16;
%
% p_std = std(p);
%
% r = find(abs(p)>factorH*p_std);
%
% p(r) = sign(p(r))*factorH*p_std;
%
%
% u1c_std = std(u1c);
%
% r = find(abs(u1c)>factor1*u1c_std);
%
% u1c(r) = sign(u1c(r))*factor1*u1c_std;
%
% u2c_std = std(u2c);
%
% r = find(abs(u2c)>factor2*u2c_std);
%
% u2c(r) = sign(u2c(r))*factor2*u2c_std;
%
% uzc_std = std(uzc);
%
% r = find(abs(uzc)>factorz*uzc_std);
%
% uzc(r) = sign(uzc(r))*factorz*uzc_std;
```

```
%bindata=[p uzc u1c u2c];  % matrix with full day/all 4 channels

% produce lofargram of desired channel

signal = p;   % allows user to choose which channel to produce
lofargram

%Fs = 200;  % sampling frequency

NFFT = 2^12;            % length of FFT

Fs = 200;  % Sampling frequency in Hz

window = hamming(NFFT);

avg = 1;                % number of averages to take

chunk = NFFT*avg;     %   number of data points needed to get averages

N = floor(length(p)/chunk);  % Total number of chunks available

lofar = ones(N,NFFT/2+1);    % initialize the matrix to make for loop
faster

x = ones(1,NFFT/2+1);


for i = 1:N                  % calculate avg power spectrum for each
chunk

   low = 1+(i-1)*chunk;
   high = i*chunk;

[x,f] = pwelch(signal(low:high),window,NFFT/2,NFFT,Fs);

lofar(i,:)=x;     % make the ith row of the lofargram the power
spectrum for that chunk

t(i) = i*chunk/Fs;   % the time corresponding to the chunk

end

lofar = 10*log10(lofar);

t = t’;

t = t/60;  % time in minutes
```

```
figure(1)

pcolor(f,t,lofar)
shading interp
title(‘LOFARgram’)
xlabel(‘Frequency (Hz)’)
ylabel(‘Time (min)’)
colorbar

% look and listen to data

answer = 1;

while answer==1;

    start_time = input(‘Input start time to look at data in minutes.’)
% start time for movie in minutes

    start_index = ceil(start_time*60*Fs)+1;

    end_time = input(‘Input end time for data in minutes.’)   % end
time for movie in minutes

    end_index = floor(end_time*60*Fs);

    signal1 = p(start_index:end_index);

    signal2 = uzc(start_index:end_index);


    [C,f]= mscohere(signal1,signal2,window,NFFT/2,NFFT,Fs);     %
[Cxy,F] = MSCOHERE(X,Y,WINDOW,NOVERLAP,NFFT,Fs)

    figure(2)

    subplot(2,1,1)
    pcolor(f,t,lofar)
    shading interp
    title(‘LOFARgram’)
    ylabel(‘Time (min)’)
    axis([0 100 start_time end_time])

    subplot(2,1,2)
    plot(f,C)
    title(‘Coherence’)
    xlabel(‘Frequency (Hz)’)
    ylabel(‘Coherence’)
    answer = input(‘Do you want to look at another sequence? 1= yes 2 =
no’)

end
```

E. READFFFAST

```
%   readFFfast
%   Read the full-field file generated by PE.

clear tl depths rangenm;

fid = fopen(‘C:\PCIMAT\archive\PE\trf\OBS10
Day18\temp0000f001.ff’);
istart = fread(fid, 1, ‘int32’);
%istart is variable to key on whether the file is STAPLE
generated vs
%PCIMAT generated.  the difference is how the information
is stored.  if
%istart > 0 the data is stored 1 record (all depths) per
range whereas if
%istart < 0 the data is stored 1 record (all ranges) per
depth
if istart < 0;
    vbRangeStart = -istart;
elseif istart > 0;
    vbDepthStart = istart;
end
vbDataStart  = fread(fid, 1, ‘int32’);
nvbRec   = fread(fid, 1, ‘int32’);
maxrange = fread(fid, 1, ‘float32’);
model    = fscanf(fid, ‘%8s’, 1);
title1   = fscanf(fid, ‘%80s’, 1);
status  = fseek(fid, 97,-1);
freq  = fread(fid, 1, ‘float32’);
fixedDepth = fread(fid, 1, ‘float32’);

if istart > 0;
    ravg = fread(fid, 1, ‘float32’);
    status  = fseek(fid, vbDepthStart-1, -1);
    ndepths = fread(fid, 1, ‘int16’);
    depths  = fread(fid, ndepths, ‘float32’);
    status  = fseek(fid, vbDataStart-1, -1);
    rangeflag = 0;
    ii = 1;
    rangenm(ii) = str2num (fscanf(fid,’%10s’,1 ));
    while ~isempty(rangeflag);
        rangenm(ii) = rangeflag;
        %       bdepth(ii) = str2num( fscanf(fid,’%10s’,
1));  9–17–2008 didn’t
        %       work need to ask Laurie why ...old readff
did work
        bdepth(ii) = fscanf(fid,’%10f’, 1);
        RuthArray = fread(fid, [2,ndepths], ‘uint8’);
        tl(ii,:) = 25.6 * RuthArray(1,:) + 0.1 *
RuthArray(2,:);
        ii = ii + 1 ;
        clear rangeflag;
        rangeflag  = str2num(fscanf(fid,’%10s’,1 ));
    end
elseif istart < 0;
    status  = fseek(fid, vbRangeStart-1, -1);
    nrange = fread(fid, 1, ‘int16’);
    rangenm  = fread(fid, nrange, ‘float32’);
    bdepth  = fread(fid, nrange, ‘float32’);
    ndepths = fread(fid, 1, ‘int16’);
    depths   = fread(fid, ndepths, ‘float32’);
    status  = fseek(fid, vbDataStart-1, -1);
    for jz = 1:ndepths;
        irec = vbDataStart + 20 + (jz-1) * nvbRec-1;
        fseek(fid, irec, -1);
        RuthArray = fread(fid, [2, nrange], ‘uint8’);
        tl(:, jz) =  25.6 * RuthArray(1,:) + 0.1 *
RuthArray(2,:);
    end
end

fclose(‘all’);
```

F. READOBS_TL

```
% File to read the TL data vs range and depth for each
frequency and then
% replace with interpolated version. There are some
peculiarities in this
% code which are designed to correct problems with the
FFfast data from the
% PC-IMAT PE output. The PC-IMAT run was done at 48.36N and
128.71W for
```

```
% depth of 2480m or 8137 ft. Frequency range was 5–80Hz.
Historical SSP
% was used for Sept 8, 2009 at 1330 local time. Bearing was
000 and max
% range 10kyds. SSP, Bathymetry, BL were “best resolution.”
Surface
% boundary was open-ocean and wind speed 20kts. Two range
vectors (19 and 20) were not
% monotically increasing, so the last range values were
deleted before
% interpolation. In this case all depth vectors were the
same for all
% frequencies.

%readFFfast
%   Read the full-field file generated by PE.

clear tl depths rangenm;

frequency = zeros(1,40);

max_depth = 1E6;

max_range = 1E6;

for n = 1:40;

    if n<=9
        filename = [‘C:\PCIMAT\archive\PE\trf\OBS10
Day18\temp0000f00’ num2str(n) ‘.ff’];
    else
        filename = [‘C:\PCIMAT\archive\PE\trf\OBS10
Day18\temp0000f0’ num2str(n) ‘.ff’];
    end

    fid = fopen(filename);
    istart = fread(fid, 1, ‘int32’);
    %istart is variable to key on whether the file is
STAPLE generated vs
    %PCIMAT generated.  the difference is how the
information is stored.  if
    %istart > 0 the data is stored 1 record (all depths)
per range whereas if
    %istart < 0 the data is stored 1 record (all ranges)
per depth
    if istart < 0;
```

```
        vbRangeStart = -istart;
         elseif istart > 0;
        vbDepthStart = istart;
    end

vbDataStart  = fread(fid, 1, ‘int32’);
nvbRec   = fread(fid, 1, ‘int32’);
maxrange = fread(fid, 1, ‘float32’);
model    = fscanf(fid, ‘%8s’, 1);
title1   = fscanf(fid, ‘%80s’, 1);
status  = fseek(fid, 97,-1);

freq  = fread(fid, 1, ‘float32’);
fixedDepth = fread(fid, 1, ‘float32’);

if istart > 0;
    ravg = fread(fid, 1, ‘float32’);
    status  = fseek(fid, vbDepthStart-1, -1);
    ndepths = fread(fid, 1, ‘int16’);
    depths  = fread(fid, ndepths, ‘float32’);
    status  = fseek(fid, vbDataStart-1, -1);
    rangeflag = 0;
    ii = 1;
    rangenm(ii) = str2num (fscanf(fid,’%10s’,1 ));
    while ~isempty(rangeflag);
        rangenm(ii) = rangeflag;
        %       bdepth(ii) = str2num( fscanf(fid,’%10s’,
1));  9–17–2008 didn’t
        %       work need to ask Laurie why ...old readff
did work
        bdepth(ii) = fscanf(fid,’%10f’, 1);
        RuthArray = fread(fid, [2,ndepths], ‘uint8’);
        tl(ii,:) = 25.6 * RuthArray(1,:) + 0.1 *
RuthArray(2,:);
        ii = ii + 1 ;
        clear rangeflag;
        rangeflag  = str2num(fscanf(fid,’%10s’,1 ));
    end
elseif istart < 0;
    status  = fseek(fid, vbRangeStart-1, -1);
    nrange = fread(fid, 1, ‘int16’);
    rangenm  = fread(fid, nrange, ‘float32’);
    bdepth  = fread(fid, nrange, ‘float32’);
    ndepths = fread(fid, 1, ‘int16’);
    depths   = fread(fid, ndepths, ‘float32’);
    status  = fseek(fid, vbDataStart-1, -1);
    for jz = 1:ndepths;
        irec = vbDataStart + 20 + (jz-1) * nvbRec-1;
        fseek(fid, irec, -1);
        RuthArray = fread(fid, [2, nrange], ‘uint8’);
        tl(:, jz) =  25.6 * RuthArray(1,:) + 0.1 *
RuthArray(2,:);
    end
end

frequency(n)=freq;

rangekyds = rangenm*2;

eval([‘rangekyds’ num2str(n) ‘ = rangekyds’]);

if max(depths) < max_depth
    max_depth = max(depths);
end

if max(rangekyds) < max_range
    max_range = max(rangekyds);
end

eval([‘tl’ num2str(n) ‘ = tl’]);

end

depths = linspace(0,max_depth,119);
rangekyds = linspace(0,max_range,150)’;

tl1 = 10.^(tl1/20);
tl1int =
interp2(depths,rangekyds1,tl1,depths,rangekyds,'linear');
tl1int = 20*log10(tl1int);
tl1 = 20*log10(tl1);

tl2 = 10.^(tl2/20);
tl2int =
interp2(depths,rangekyds2,tl2,depths,rangekyds,'linear');
tl2int = 20*log10(tl2int);
tl2 = 20*log10(tl2);

tl3 = 10.^(tl3/20);
tl3int =
interp2(depths,rangekyds3,tl3,depths,rangekyds,'linear');
tl3int = 20*log10(tl3int);
```

```
tl3 = 20*log10(tl3);

tl4 = 10.^(tl4/20);
tl4int =
interp2(depths,rangekyds4,tl4,depths,rangekyds,'linear');
tl4int = 20*log10(tl4int);
tl4 = 20*log10(tl4);

tl5 = 10.^(tl5/20);
tl5int =
interp2(depths,rangekyds5,tl5,depths,rangekyds,'linear');
tl5int = 20*log10(tl5int);
tl5 = 20*log10(tl5);

tl6 = 10.^(tl6/20);
tl6int =
interp2(depths,rangekyds6,tl6,depths,rangekyds,'linear');
tl6int = 20*log10(tl6int);
tl6 = 20*log10(tl6);

tl7 = 10.^(tl7/20);
tl7int =
interp2(depths,rangekyds7,tl7,depths,rangekyds,'linear');
tl7int = 20*log10(tl7int);
tl7 = 20*log10(tl7);

tl8 = 10.^(tl8/20);
tl8int =
interp2(depths,rangekyds8,tl8,depths,rangekyds,'linear');
tl8int = 20*log10(tl8int);
tl8 = 20*log10(tl8);

tl9 = 10.^(tl9/20);
tl9int =
interp2(depths,rangekyds9,tl9,depths,rangekyds,'linear');
tl9int = 20*log10(tl9int);
tl9 = 20*log10(tl9);

tl10 = 10.^(tl10/20);
tl10int =
interp2(depths,rangekyds10,tl10,depths,rangekyds,'linear');
tl10int = 20*log10(tl10int);
tl10 = 20*log10(tl10);

tl11 = 10.^(tl11/20);
```

```
tl11int =
interp2(depths,rangekyds11,tl11,depths,rangekyds,’linear’);
tl11int = 20*log10(tl11int);
tl11 = 20*log10(tl11);

tl12 = 10.^(tl12/20);
tl12int =
interp2(depths,rangekyds12,tl12,depths,rangekyds,’linear’);
tl12int = 20*log10(tl12int);
tl12 = 20*log10(tl12);

tl13 = 10.^(tl13/20);
tl13int =
interp2(depths,rangekyds13,tl13,depths,rangekyds,’linear’);
tl13int = 20*log10(tl13int);
tl13 = 20*log10(tl13);

tl14 = 10.^(tl14/20);
tl14int =
interp2(depths,rangekyds14,tl14,depths,rangekyds,’linear’);
tl14int = 20*log10(tl14int);
tl14 = 20*log10(tl14);

tl15 = 10.^(tl15/20);
tl15int =
interp2(depths,rangekyds15,tl15,depths,rangekyds,’linear’);
tl15int = 20*log10(tl15int);
tl15 = 20*log10(tl15);

tl16 = 10.^(tl16/20);
tl16int =
interp2(depths,rangekyds16,tl16,depths,rangekyds,’linear’);
tl16int = 20*log10(tl16int);
tl16 = 20*log10(tl16);

tl17 = 10.^(tl17/20);
tl17int =
interp2(depths,rangekyds17,tl17,depths,rangekyds,’linear’);
tl17int = 20*log10(tl17int);
tl17 = 20*log10(tl17);

tl18 = 10.^(tl18/20);
tl18int =
interp2(depths,rangekyds18,tl18,depths,rangekyds,’linear’);
tl18int = 20*log10(tl18int);
tl18 = 20*log10(tl18);
```

```
tl19 = 10.^(tl19/20);
tl19int =
interp2(depths,rangekyds19(1:38),tl19(1:38,:),depths,rangek
yds,’linear’);
tl19int = 20*log10(tl19int);
tl19 = 20*log10(tl19);

tl20 = 10.^(tl20/20);
tl20int =
interp2(depths,rangekyds20(1:38),tl20(1:38,:),depths,rangek
yds,’linear’);
tl20int = 20*log10(tl20int);
tl20 = 20*log10(tl20);

tl21 = 10.^(tl21/20);
tl21int =
interp2(depths,rangekyds21,tl21,depths,rangekyds,’linear’);
tl21int = 20*log10(tl21int);
tl21 = 20*log10(tl21);

tl22 = 10.^(tl22/20);
tl22int =
interp2(depths,rangekyds22,tl22,depths,rangekyds,’linear’);
tl22int = 20*log10(tl22int);
tl22 = 20*log10(tl22);

tl23 = 10.^(tl23/20);
tl23int =
interp2(depths,rangekyds23,tl23,depths,rangekyds,’linear’);
tl23int = 20*log10(tl23int);
tl23 = 20*log10(tl23);

tl24 = 10.^(tl24/20);
tl24int =
interp2(depths,rangekyds24,tl24,depths,rangekyds,’linear’);
tl24int = 20*log10(tl24int);
tl24 = 20*log10(tl24);

tl25 = 10.^(tl25/20);
tl25int =
interp2(depths,rangekyds25,tl25,depths,rangekyds,’linear’);
tl25int = 20*log10(tl25int);
tl25 = 20*log10(tl25);

tl26 = 10.^(tl26/20);
```

```
tl26int =
interp2(depths,rangekyds26,tl26,depths,rangekyds,'linear');
tl26int = 20*log10(tl26int);
tl26 = 20*log10(tl26);

tl27 = 10.^(tl27/20);
tl27int =
interp2(depths,rangekyds27,tl27,depths,rangekyds,'linear');
tl27int = 20*log10(tl27int);
tl27 = 20*log10(tl27);

tl28 = 10.^(tl28/20);
tl28int =
interp2(depths,rangekyds28,tl28,depths,rangekyds,'linear');
tl28int = 20*log10(tl28int);
tl28 = 20*log10(tl28);

tl29 = 10.^(tl29/20);
tl29int =
interp2(depths,rangekyds29,tl29,depths,rangekyds,'linear');
tl29int = 20*log10(tl29int);
tl29 = 20*log10(tl29);

tl30 = 10.^(tl30/20);
tl30int =
interp2(depths,rangekyds30,tl30,depths,rangekyds,'linear');
tl30int = 20*log10(tl30int);
tl30 = 20*log10(tl30);

tl31 = 10.^(tl31/20);
tl31int =
interp2(depths,rangekyds31,tl31,depths,rangekyds,'linear');
tl31int = 20*log10(tl31int);
tl31 = 20*log10(tl31);

tl32 = 10.^(tl32/20);
tl32int =
interp2(depths,rangekyds32,tl32,depths,rangekyds,'linear');
tl32int = 20*log10(tl32int);
tl32 = 20*log10(tl32);

tl33 = 10.^(tl33/20);
tl33int =
interp2(depths,rangekyds33,tl33,depths,rangekyds,'linear');
tl33int = 20*log10(tl33int);
tl33 = 20*log10(tl33);
```

```
tl34 = 10.^(tl34/20);
tl34int =
interp2(depths,rangekyds34,tl34,depths,rangekyds,’linear’);
tl34int = 20*log10(tl34int);
tl34 = 20*log10(tl34);

tl35 = 10.^(tl35/20);
tl35int =
interp2(depths,rangekyds35,tl35,depths,rangekyds,’linear’);
tl35int = 20*log10(tl35int);
tl35 = 20*log10(tl35);

tl36 = 10.^(tl36/20);
tl36int =
interp2(depths,rangekyds36,tl36,depths,rangekyds,’linear’);
tl36int = 20*log10(tl36int);
tl36 = 20*log10(tl36);

tl37 = 10.^(tl37/20);
tl37int =
interp2(depths,rangekyds37,tl37,depths,rangekyds,’linear’);
tl37int = 20*log10(tl37int);
tl37 = 20*log10(tl37);

tl38 = 10.^(tl38/20);
tl38int =
interp2(depths,rangekyds38,tl38,depths,rangekyds,’linear’);
tl38int = 20*log10(tl38int);
tl38 = 20*log10(tl38);

tl39 = 10.^(tl39/20);
tl39int =
interp2(depths,rangekyds39,tl39,depths,rangekyds,’linear’);
tl39int = 20*log10(tl39int);
tl39 = 20*log10(tl39);

tl40 = 10.^(tl40/20);
tl40int =
interp2(depths,rangekyds40,tl40,depths,rangekyds,’linear’);
tl40int = 20*log10(tl40int);
tl40 = 20*log10(tl40);

figure(1)  % check to make sure interpolated values match
originals
subplot(2,1,1)
```

```
imagesc(rangekyds1,depths,tl1’)
colorbar
xlabel(‘Range (kyds)’); ylabel(‘Depth (ft)’)
title([‘TL at Frequency of ‘ num2str(frequency(1)) ‘ Hz’])

subplot(2,1,2)
imagesc(rangekyds,depths,tl1int’)
colorbar
xlabel(‘Range (kyds)’); ylabel(‘Depth (ft)’)
title([‘Interpolated TL at Frequency of ‘
num2str(frequency(1)) ‘ Hz’])

figure(40)
subplot(2,1,1)
imagesc(rangekyds40,depths,tl40’)
colorbar
xlabel(‘Range (kyds)’); ylabel(‘Depth (ft)’)
title([‘TL at Frequency of ‘ num2str(frequency(40)) ‘ Hz’])

subplot(2,1,2)
imagesc(rangekyds,depths,tl40int’)
colorbar
xlabel(‘Range (kyds)’); ylabel(‘Depth (ft)’)
title([‘Interpolated TL at Frequency of ‘
num2str(frequency(40)) ‘ Hz’])

fclose(‘all’);

%  Now put together all the interpolated TL matrices into a
3D matrix
% The rows are ranges, columns are depths, and height is
frequency.
% So to pick out a TL for a given range (which corresponds
to time)
% and a given depth, you take TL(range_row, depth_column,:)

TL = zeros(150,119,40); %(range, depth, freq)

for n = 1:40;

eval([‘TL(:,:,’ num2str(n) ‘) = tl’ num2str(n) ‘int’]);

end
```

G. LOFAR

```
range=1; %kyds
rangei=round((range+.066)/.066); %index
freq=60; %Hz
freqi=round(log(freq/4.655)/.071); %index
depth=0:30:300; %ft
for m=1:11
    depthi=round((depth(m)+88.5)/88.5); %index
CPA=0; %kyds
v=15; %kts

%TL(range,depth,freq)
TLI=TL(:,depthi,:); %singles out set depth
tlrange=squeeze(TLI);

if m==1
figure(m)
imagesc(frequency,rangekyds,tlrange)
ylabel(‘range (kyds)’)
xlabel(‘freq (Hz)’)
title([‘Source depth of ‘ num2str(depth(m)) ‘ ft’])
else
   figure(2*m-1)
imagesc(frequency,rangekyds,tlrange)
ylabel(‘range (kyds)’)
xlabel(‘freq (Hz)’)
title([‘Source depth of ‘ num2str(depth(m)) ‘ ft’])
t=sqrt(rangekyds.^2-CPA^2)/(v*2/3600);
end

tltime=zeros(150,40);
for n=1:150
    tltime(n,:)=tlrange(n,:).*real(t(n));
end

if m==1
figure(m+1)
imagesc(frequency,t,tltime)
ylabel(‘time (sec)’)
xlabel(‘freq (Hz)’)
title([‘Source depth of ‘ num2str(depth(m)) ‘ ft’])
else
    figure(2*(m))
imagesc(frequency,t,tltime)
ylabel(‘time (sec)’)
```

```
xlabel(‘freq (Hz)’)
title([‘Source depth of ‘ num2str(depth(m)) ‘ ft’])
end

end
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] J. Scobo, “Ocean observing systems,” M.S. thesis, Naval Postgraduate School, Monterey, CA, 2012.

[2] R. Rayner. (n.d.). “The global ocean observing system.” Available: http://www.lse.ac.uk/CATS/Events/AnnivDocs/Rayner_talk.pdf

[3] Woods Hole Oceanographic Institute. (2013 July 24). “Ocean-bottom seismometer.” Available: http://www.whoi.edu/instruments/viewInstrument.do?id=10347

[4] Woods Hole Science Center. (2013, July 24). “What is an ocean bottom seismometer?” Available: http://woodshole.er.usgs.gov/operations/obs/whatobs.html

[5] Woods Hole Science Center. (2013, July 24). “OBS image.” Available: http://woodshole.er.usgs.gov/operations/obs/Images/NZ96/obs_nz.gif

[6] Incorporated Research Institute for Seismology. (2013, July 24). “IRIS DMC MetaData Aggregator.” Available: http://www.iris.edu/mda/YN/OBS10?timewindow=2009–2009

[7] Quanterra, (2013, August 12). “Quanterra Q330.” Available: http://www.q330.com

[8] Ocean Bottom Seismograph Instrument Pool. (2013, July 24). “OBS’s for landlubbers.” Available: http://www.obsip.org

[9] High Tech INC. (2013, August 12). “HTI-90-U.” Available: http://www.hightechincusa.com/90_U.html

[10] Heat, Light & Sound Research, Inc. (2013, August 6). “Vector sensors.” Available: http://www.hlsresearch.com/research/vectorsensors.html

[11] B. A. Cray and A. H. Nuttall, “Directivity factors for linear arrays of velocity sensors,” *J. Acoust. Soc. Am.* vol. 110 (1), pp. 324–331, 2001.

[12] D. Kapolka, J. V. Caulk, and K. B. Smith, “Experimental and theoretical performance of a hybrid vector sensor/microphone acoustic array,” *Proceedings of the 10th European Conference on Underwater Acoustics*, Istanbul, Turkey, 5–9 July, 2010.

[13] C. W. Ng, “Experimental tracking of aerial targets using the Microflown sensor,” M.S. thesis, Naval Postgraduate School, Monterey, CA, 2012.

[14] B. T. R. Lewis and J. McClain, “Converted shear waves as seen by ocean bottom seismometers and surface buoys,” *Bulletin of the Seismological Soc. of Am.,* vol. 67, no. 5, pp. 1291–1302, October 1977.

[15] F. K. Duennebier and G. H. Sutton, “Fidelity of ocean bottom seismic observations,” *Marine Geophysical Researches*, vol. 17, pp. 535–555, 1995.

[16] J. C. Osler and D. M. F. Chapman, “Quantifying the interaction of an ocean bottom seismometer with the seabed,” *J. of Geophysical Research*, vol. 103, no. B5, pp. 9879–9894, May 1998.

[17] K. B. Smith, “Effects of shear waves on boundary-coupled vector sensors,” *J. Acoust. Soc. Am.*, vol. 124 (6), pp. 3464–3470, December 2008.

[18] Wikipedia. (2013, July 29). “Lloyd’s mirror.” Available: http://en.wikipedia.org/wiki/Lloyd’s_mirror.

[19] R.J. Urick, *Principles of Underwater Sound*, rev. 1st ed. McGraw-Hill, 1975, pp120–123.

[20] R. Hudson, “A horizonal range vs.depth solution of sound source position under general sound velocity conditions using the Lloyd’s mirror interference pattern,” M.S. thesis, Naval Postgraduate School, Monterey, CA, 1983.

[21] S. D. Chuprov, “Interference structure of a sound field in a layered ocean,” in *Ocean Acoustics. Current State,* L. M. Brekhovskikh and I. B Andreevoi, Eds. Moscow: Nauka, 1982.

[22] L. Brekhovskikh and Y. Lysanov, Y. P. *Fundamentals of Ocean Acoustics*, 3rd Edition. New York: Springer-Verlag, 2003.

[23] G. D`Spain and W. Kuperman, “Application of waveguide invariants to analysis of spectrograms from shallow water environments that vary in range and azimuth.” *J. Acoust. Soc. Am.*, vol.106(5), pp. 2454–2468, November 1999.

[24] Thode, A. M., “Source ranging with minimal environmental information using a virtual receiver and waveguide invariant theory,” *J. Acoust. Soc. Am.*, vol. 108(4), pp. 1582–1594, October 2000.

[25] S. Narasimhan and J. L. Krolik, “Fundamental limits on acoustic source range estimation performance in uncertain ocean channels,” *J. Acoust. Soc. Am.*, vol. 97(1), pp. 215–226, January 1995.

[26] L. M. Zurk and B. H. Tracey, “Depth-shifting of shallow water guide source observations,” *J. Acoust. Soc. Am.*, vol. 118(4), pp. 2224–2233, October 2005.

[27] D. Rouseff and C. V. Leigh, “Using the waveguide invariant to analyze lofargrams,” in *Oceans ‘02 MTS/IEEE*, 2002, vol. 4, pp. 2239–2243.

[28] D. Rouseff and L. M. Zurk, “Striation-based beamforming for estimating the waveguide invariant with passive sonar,” *J. Acoust. Soc. Am.*, vol. 130(2), pp. 76–81, August 2011.

[29] S. Lee and N. C. Makris, “The array invariant,” *J. Acoust. Soc. Am.*, vol. 119(1), pp. 336–351, January 2006.

[30] L. M. Zurk and D. Rouseff, “Striation-based beamforming for active sonar with a horizontal line array,” *J. Acoust. Soc. Am.*, vol. 132(4), October 2012.

[31] L. A. Brooks *et al.*, “Techniques for extraction of the waveguide invariant from interference patterns in spectrograms,” *Proceedings of ACOUSTICS 2006*, 20–22 November 2006.

[32] D. Kapolka *et al.*, “Equivalence of the waveguide invariant and two path ray theory methods for range prediction based on Lloyd’s mirror patterns,” *Proc. of Meetings on Acoustics*, vol. 4, 155th Meeting Acoustical Society of America, 2008.

[33] D. Kapolka,”Underwater Acoustics for Naval Applications,” class notes for PH3452, Naval Postgraduate School, Winter 2013.

[34] R. Keenan (private communication), April 2013

[35] R. Keenan (private communication), September 2013

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California